

第二十二年度
ソフトウェア・メンテナンス研究会
報告書

2013年9月
ソフトウェア・メンテナンス研究会
<http://www.serc-j.jp/>

はじめに

20 数年前にこの研究会がスタートしたころ、世界の主流は汎用コンピュータの上で動く大規模アプリケーション・システムであり、研究会のメンバーもそうしたシステムを保守する仕事に携わっている人たちが多かったと記憶する。

それから四半世紀の時間が経過するうちに、われわれを取り巻く社会の環境はめまぐるしく変化して現在にいたっている。インターネットがほぼ全世界をカバーし、いくつものクラウド・アプリケーションが台頭しはじめ、モバイル・デバイスが人びとの日常生活を支えるツールとして利用される一方で、汎用マシンの上ではいまだにレガシーな旧世代のシステムが稼働しているという混沌とした状況である。

かつて M..M..Lehman 先生が予言したように、現実世界のアプリケーションに組み込まれたソフトウェア・システムは、いったん開発が完了した後も成長・進化をし続ける。その進化プロセスは、マルチ・エージェント、マルチ・レベル、マルチ・ループのフィードバック・システムであり、それを保守・維持して行く仕事の重要性は以前にもまして大きくなってきている。

にもかかわらず、世の中での関心は常に新しいシステムの技術開発に焦点が当てられており、システムの保守や進化に陽が当たることは少ない。この研究会は、そうした状況を打開すべくこれまで活動を展開してきた。これからも地道にその努力を続けて行かなければならないだろう。

2013 年秋

ソフトウェア・メンテナンス研究会

代 表 幹 事 岸田孝一

目次

第二十二年度研究員名簿	4
第二十二年度活動報告	5
1. フォーラム資料	
資料1-1 2013年2月22日 消費税率変更対応。本当に大丈夫？	7
2. 作業グループ報告書	
保守技術者の教育	53
保守作業改善の基盤技術調査	68
10年後のソフトウェア保守を考える	94
SERCの考える保守とは	143
3. まとめ	
幹事よりひとこと	208

第二十二年度研究員名簿

組織名	氏名
(株)NTT データ CCS	馬場 辰男
(株)SRA	古石 ゆみ 方 学芬 岸田 孝一
(株)バイトルヒクマ	弘中 茂樹
(株)精密形状処理研究所	長谷川 亨
(株)中電シーティーアイ	諸岡 隆司
(株)日立ソリューションズ	木部 俊之 鈴木 勝彦 高橋 宏志 中山 優紀 松本 道春
(財)経済調査会	押野 智樹
(有)ウィルビーネットワーク	松尾 好博
NARA コンサルティング	奈良 隆正
アイエックス・ナレッジ(株)	井瀬 英晶 岡田 浩 田中 一夫
システム企画研修(株)	上野 則男
東芝ソリューション(株)	沼田 恵助 川上 康史 佐井 由美子 野口 大輔 増井 和也
個人研究員	石川 雅彦 伊藤 順一 江尻 武志 大島 道夫 小林 允 塩谷 和範 福崎 哲郎 福島 茂雄 丸山 陽一 峰村 圭介 高橋 芳広 三輪 東

第二十二年度活動報告

ソフトウェア・メンテナンス・シンポジウム

テーマ:ソフトウェアの進化

日時:2012年10月15日(金)

基調講演:「ソフトウェア進化の研究」

立命館大学 丸山 勝久 教授

定例活動

「2. 研究グループ報告書」参照

葉山集中研修

日時:2012年11月30日(金)～ 12月1日(土)

場所:逗子市 葉山研修センター

内容:

<初日>

13:30- 開会宣言:代表幹事 岸田孝一 様

13:35- 会員拡大策:全員での討議

16:00- 基調講演:ソフトウェア保守に影響する要因の分析

東洋大学 角田 雅照 氏

18:00- 情報交換会 なじま

<2日目>

9:00- グループでの計画立案

12:30- 昼食 菊水亭

13:00- グループでの計画立案

15:00- 各グループ発表

14:00 解散

SERC Forum

2013年2月22日(金) 13:30-16:50 消費税率変更対応。本当に大丈夫？

協賛

派生開発カンファレンス2013 <http://www.xddp.jp/conference2013.shtml>

ソフトウェア・シンポジウム2013 <http://sea.jp/ss2013/>

1. 講 演 資 料

SERC フォーラム

消費税率変更対応。本当に大丈夫？

主催 ソフトウェア・メンテナンス研究会(SERC)

<http://www.serc-j.jp/>

ソフトウェア・メンテナンス研究会は 1990 年の設立以来、国内唯一のソフトウェア進化・保守に関する研究グループとして、ソフトウェア保守の諸問題に焦点を当て、さまざまな観点から解決策を探ってまいりました。

SERC フォーラム、今年度第 1 回は 2014 年度に実施される消費税改正に伴うソフトウェア保守の課題を取り上げます。昨年暮れの政権交代後、行き過ぎた円高の是正が行われ、株価も上昇に転じ、消費税率の引き上げの環境が整いつつありますが、皆様のソフトウェアやシステムの対応の準備は如何でしょうか？

去る 1997 年にも消費税率の引き上げが行われましたが、今や 15 年が経ち、当時の資料は散逸し、当時を知る技術者も少なくなったせいか、必要な作業が見えず単に税率の定数を 5 % から 8 % に替えれば良いとの話をする人もいます。消費税率引き上げ対応の修正作業見積もりや体制確保を軽視されている現場や経営者はいないでしょうか？しかしながら、15 年前の改正時には旧消費税率と新消費税率とが混在する時期（経過措置）があり、今回の改正時も同様な措置が考えられます。さらに、10 % への変更時には軽減税率の採用が検討されており、さらに大きな影響が見込まれています。

そこで、当研究会では「消費税率変更対応。本当に大丈夫？」と題しまして、消費税引き上げ対応作業の課題を認識するためのフォーラムを開催することにしました。

今回は、原田税務会計事務所殿のご協力により税務の専門家から、消費税率変更概要および業務やシステムに与えるインパクトを説明して頂きます。

さらに、1997 年の消費税率変更対応の報告、今回の消費税率変更の課題をソフトウェア保守の観点からの報告など、盛りだくさんな内容になっております。

《開催要領》

●日時：2013 年 2 月 22 日（金）13:30-16:50（受付 13:00-）

●プログラム

13:00 - 13:30 受付

13:30 - 13:35 開会の挨拶

13:35 - 15:00 基調講演 「消費税の改正点と実務における重要ポイント」…………… Pg3
原田税務会計事務所 税理士 藤森 康彦氏

15:00 - 15:20 休息

15:20 - 16:00 研究会報告「消費税率に対する取組み～あるシステムでの事例」…………… Pg29
SERC 研究員 弘中 茂樹氏

16:00 - 16:40 研究会報告「消費税率変更の課題－ソフトウェア保守の観点から－」…………… Pg39
SERC 幹事 増井 和也氏

16:40 - 16:50 クロージング

消費税の改正点と実務における 重要ポイント

1. 概要

消費一般に対して広く公平に負担を求めるため、税率3%で平成元年4月1日から適用されました。

平成9年4月から消費税率4%とされたほか、消費税と課税の対象を同一にし、消費税率換算で1%の地方消費税が創設され、消費税とあわせて賦課徴収することとされました。

我が国において今後ますます少子高齢化が進展することを前提として、従来から社会保障改革の必要性とともに財政の健全化を図るための税制改革が検討されてきました。

この度の消費税率引上げを中心とした消費税法等改正までの経緯は、次のとおりとなります。

平成22年12月14日に「社会保障改革の推進について」閣議決定。

平成24年1月6日に「社会保障・税一体改革素案」が閣議報告され、

同年2月17日に「社会保障・税一体改革大綱」が閣議決定

この「社会保障・税一体改革大綱」を踏まえて立案された「社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための消費税等の一部を改正する等の法律案」が平成24年3月30日に国会に提出されました。

国会における審議の過程で衆議院において法案の一部修正が行われ、その後参議院の審議を経て、平成24年8月10日に「社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための消費税等の一部を改正する等の法律」が成立、同年8月22日に交付されました。

なお、この法律改正に併せて「社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための地方税及び地方交付税法の一部を改正する法律」も成立し、交付されています。

2. 消費税の仕組み

- ① 原則として全ての財貨・サービスの販売の国内における販売、提供を課税の対象とする。
- ② 生産、流通、販売などの全段階において、他の事業者や消費者に財貨・サービスの提供を行う事業者を納税義務者とし、その売上に対して課税を行う。
- ③ 事業者は、売上に係る税額から仕入れに係る税額を控除し、その差引き税額を納付する（控除不足額を還付する。）。
- ④ 事業者に課せられる税相当額は、財貨、サービスの販売価格に織り込まれ、最終的に消費者が負担することを予定されている。
- ⑤ 輸入取引については、保税地域から貨物を引き取る者を納税義務者として課税、輸出取引については売上に課税を行わず、仕入税額控除の不足額の還付を行う。

平成24年度一般会計予算の概要

平成24年度一般会計予算は約90兆円です。このうち歳出について見ると、国債の元利払いに充てられる費用(国債費)と地方交付税交付金等と社会保障関係費で、歳出全体の約7割を占めています。

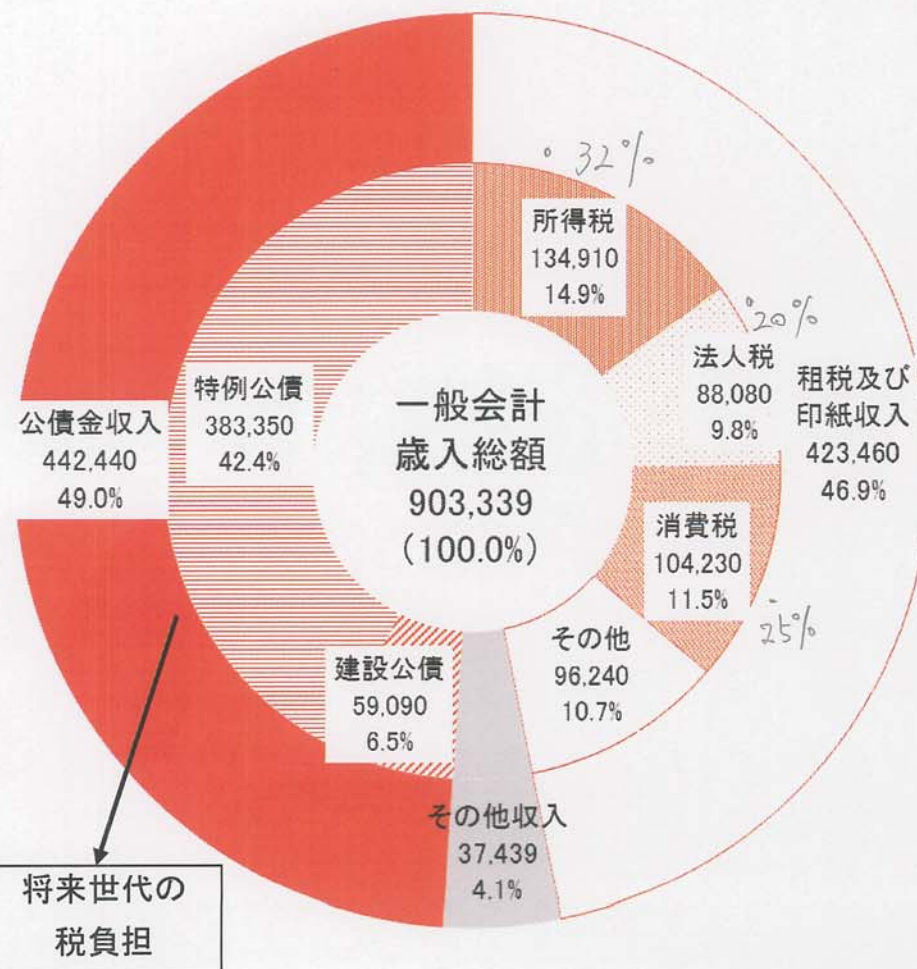
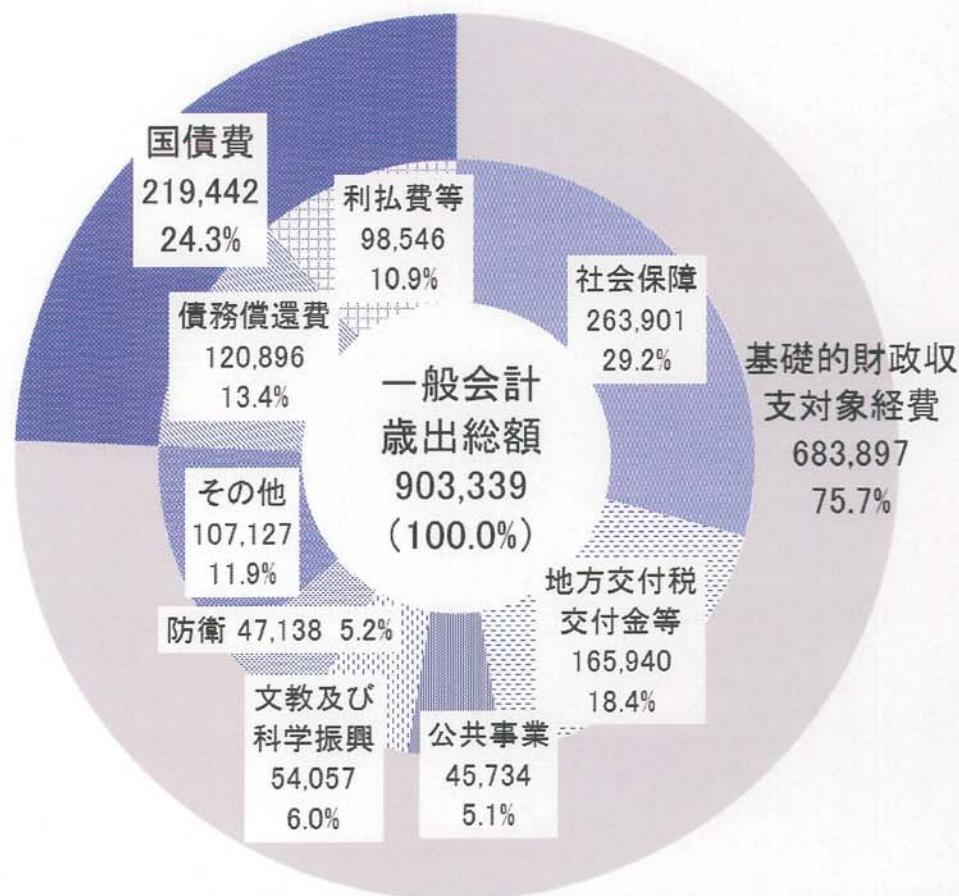
平成24年度一般会計予算における歳入のうち、税収でまかなわれているのは5割に満たず、5割弱は将来世代の負担となる借金(公債金収入)に依存しています。

歳出

(単位:億円、%)

歳入

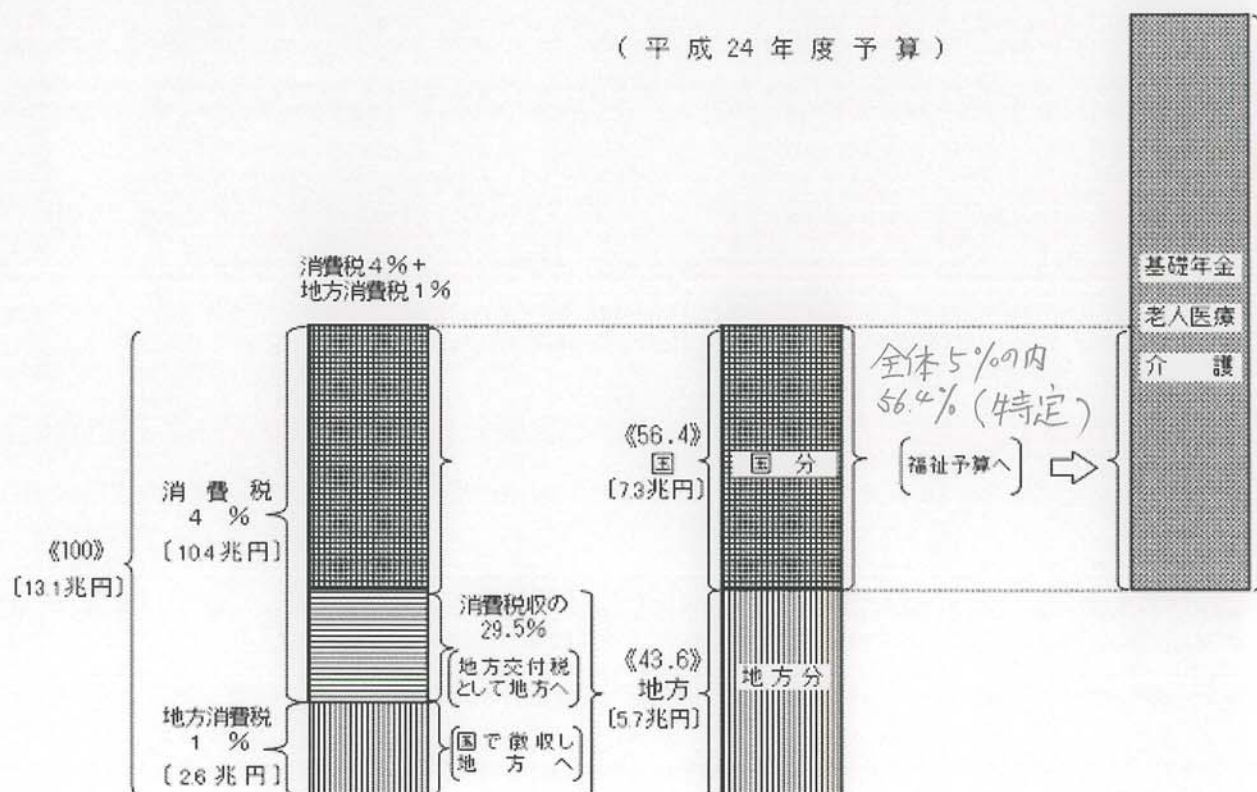
2



消費税の使途

- 消費税の収入が充てられる経費(地方交付税交付金を除く)の範囲(基礎年金、老人医療、介護)を予算決定(平成11年度予算～)

(平成24年度予算)



〒100-8940 東京都千代田区霞が関3-1-1 Tel(代表)03-3581-4111(9時～18時30分)

Copyright(C) 財務省



1. 消費税の改正事項

(2) 社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための消費税法の一部を改正する等の法律による改正事項

① 資産の譲渡等に係る消費税率引上げ改正

第2条 平成26年4月1日以後に行われる課税資産の譲渡等について、消費税率を4%から**6.3%**に引き上げる
(地方消費税1.7%と合わせて**8%**)

第3条 平成27年10月1日以後に行われる課税資産の譲渡等について、消費税率を6.3%から**7.8%**に引き上げる
(地方消費税2.2%と合わせて**10%**)

施行日前後の取引に係る税率の適用関係

1. 施行日前に売買契約した商品が施行日後に引き渡された場合の税率
8%の税率は、施行日（平成26年4月1日）以後に国内において事業者が行う課税資産の譲渡等に対して適用される。
課税資産の譲渡は、消費の引渡しがあった日。
2. 予約を受けた商品の納品が遅れた場合
自動車の駆け込み需要により、法定の手続き等の遅れにより納品が施行日以後になった場合。
3. 施行日前に予約を受けて施行日以後に役務の提供を行う場合
利用者の大半が予約をした上での来店となる高級レストラン、予約制の美容院、ホテル、旅館、宴会場
4. 施行日を含む1年間の役務提供を行うものの課税関係
1年間のコピー機メンテナンス契約
代理店契約に係る代理店手数料
施行日前に年間契約して1年分をまとめて授受する場合。
5. レジスター、自動販売機、自動発売機等の税率及び価格等の切替時期
6. 施行日以後の取引について5%の税率を適用して代金を受領した場合
レジスター等の設定が間に合わず、5%の税率で代金の清算をした場合
7. 未成工事支出金として経理した金額に基づき控除税額を計算する場合の税率



1. 消費税の改正事項

(2) 社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための消費税法の一部を改正する等の法律による改正事項

② 消費税の改正に伴う主な経過措置の規定

改正消費税率は、原則として、平成26年4月1日以後に国内において事業者が行う課税資産の譲渡等について適用されますが、**次の経過措置は、平成26年4月1日以後に事業者が行う課税資産の譲渡等であっても旧消費税率が適用されます。**

- イ 旅客運賃等に関する経過措置
- ロ 電気料金等に関する経過措置
- ハ 長期請負工事等に関する経過措置
- ニ 資産の長期貸付け（リース契約等）に関する経過措置
- ホ 事前に分割で支払われる特定の役務の提供に関する経過措置

② 消費税の改正に伴う主な経過措置の規定

改正消費税率は、原則として、平成26年4月1日以後に国内において事業者が行う課税資産の譲渡等について適用されますが、次の経過措置は、平成26年4月1日以後に事業者が行う課税資産の譲渡等であっても旧消費税率が適用されます。

イ 旅客運賃等に関する経過措置

平成26年4月1日以後に行う旅客運送サービス等の提供で、同日前にその対価を領収している場合

ロ 電気料金等に関する経過措置

継続的な供給契約に基づき、平成26年4月1日前から供給される電気等の料金で、同日から1月以内に検針等に基づき料金が確定するもの

ハ 長期請負工事等に関する経過措置

平成25年10月1日前に締結した長期請負工事等で、平成26年4月1日以後に引渡し等を行う場合

ニ 資産の長期貸付け(リース契約等)に関する経過措置

平成25年10月1日前に締結した契約に基づき、平成26年4月1日前から同日以後引き続き貸付けを行っている場合において、料金変更不可等の一定要件に該当するとき

ホ 事前に分割で支払われる特定の役務の提供に関する経過措置

あらかじめ時期を定めることのできない役務の提供で、平成25年10月1日前に対価を分割して支払う契約を締結したもの(冠婚葬祭等)で、当該役務の提供が平成26年4月1日以後に行われる場合



1. 消費税の改正事項

(2) 社会保障の安定財源の確保等を図る税制の抜本的な改革を行うための消費税法の一部を改正する等の法律による改正事項

③ 基準期間のない資本金 1 千万円未満の新設法人の免税点制度の見直し

平成26年4月1日以後に設立される資本金 1 千万円未満の新設法人について、課税売上高が 5 億円超の事業者等がグループで 50% 超出資し設立された法人については、**設立当初の 2 年間、事業者免税点制度が適用されなくなります。**

④ 中間申告義務のない事業者の任意の中間申告制度の創設

直前の課税期間の確定消費税の年税額が 48 万円以下（地方消費税を含めると 61.52 万円以下）の事業者は中間申告義務がありませんが、中間申告を行う意思を有する事業者については、届け出をすることにより**任意の中間申告が可能となります。**

この任意中間申告は、平成26年4月1日以後開始する課税期間に係るものについて適用できます。



1. 消費税の改正事項

(3) 平成23年度改正のうち平成24年4月1日以後適用される重要事項

① 全額仕入税額控除（95%ルール）の改正

- 1) 平成24年4月1日以後に開始する課税期間から課税売上割合が95%以上100%未満であっても、**その課税期間の課税売上高が5億円を超える事業者は「95%ルール」を適用できなくなっています。**
- 2) 個人事業者は、通常は平成25年分から適用されますが、**課税期間特例を選択適用している事業者は、平成24年4月1日以後に開始する課税期間から適用されています。**
- 3) 課税期間特例を選択適用している場合の課税売上高5億円超か以下かの判断は、その課税期間の課税売上高をその課税期間の月数で除し、これに12を乗じて算出した金額（**年換算した金額**）により判定します。
- 4) **課税売上割合が95%未満の事業者又は課税売上高が5億円を超える事業者は、仕入控除税額の計算を個別対応方式若しくは一括比例配分方式のいずれかにより行うことになります。**



1. 消費税の改正事項

(3) 平成23年度改正のうち平成24年4月1日以後適用される重要事項

② 「消費税の還付申告に関する明細書」の添付義務化

- 1) 平成24年4月1日以後に提出する控除不足還付税額のある還付申告書から「消費税の還付申告に関する明細書（個人事業者用）」の添付が義務付けられました。
- 2) これまで還付申告書への添付を協力要請されていた「仕入控除税額に関する明細書」の記載事項に加え、課税資産の譲渡や輸出取引に係る項目について記載することになります。
- 3) 中間納付還付税額のための還付申告書又は控除不足還付税額のない申告書には添付する必要はありません。

消費税の還付申告に関する明細書 (個人事業者用)

課税期間	・ ・ ・ ・ ・	住 所	
		氏 名	

1 還付申告となった主な理由 (該当する事項に○印を付してください。)

輸出等の免税取引の割合が高い	その他	[]
設備投資 (高額な固定資産の購入等)		

2 課税売上げ等に係る事項

(1) 主な課税資産の譲渡等 (取引金額が100万円以上の取引先を上位5番目まで記載してください。)

資 産 の 種 類 等	譲 渡 年 月 日 等	取 引 金 額 等 (税込・税抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	・ ・	円		
	・ ・			
	・ ・			
	・ ・			
	・ ・			

※ 継続的な取引先については、当課税期間中の取引金額の合計額を記載し、譲渡年月日等欄には「継続」と記載してください。輸出取引等は(2)に記載してください。

(2) 主な輸出取引等の明細 (取引金額総額の上位5番目まで記載してください。)

取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)	取 引 金 額	主 な 取 引 商 品 等	所 轄 税 関 (支 署) 名
		円		

輸 出 取 引 等 に 利 用 す る	主 な 金 融 機 関	銀 行 金 庫 ・ 組 合 農 協 ・ 漁 協		本 店 ・ 支 店 出 張 所 本 所 ・ 支 所	
		預 金	口 座 番 号		
	主 な 通 関 業 者	氏 名 (名 称)			
		住 所 (所 在 地)			

3 課税仕入れに係る事項

(1) 仕入金額等の明細

区 分		① 決 算 額 (税込・税抜)	② 左のうち課税仕入 れにならないもの	(①-②) 課税仕入高
事業所得	仕 入 金 額 (製品製造原価)	①	円	円
	必 要 経 費	②		
	固定資産等の 取 得 価 額	③		
	小 計 (①+②+③)	④		
不動産所得	必 要 経 費	⑤		
	固定資産等の 取 得 価 額	⑥		
	小 計 (⑤+⑥)	⑦		
所得	仕 入 金 額	⑧		
	必 要 経 費	⑨		
	固定資産等の 取 得 価 額	⑩		
	小 計 (⑧+⑨+⑩)	⑪		
課税仕入高の合計額		⑫	④、⑦、⑪の合計額を記載してください。	
課税仕入れ等の税額の合計額		⑬	⑫の金額に対する消費税額	

(2) 主な棚卸資産・原材料等の取得（取引金額が100万円以上の取引先を上位5番目まで記載してください。）

資 産 の 種 類 等	取 得 年 月 日 等	取 引 金 額 等 (税込・税抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	・ ・	円		
	・ ・			
	・ ・			
	・ ・			
	・ ・			

※ 継続的な取引先については、当課税期間中の取引金額の合計額を記載し、取得年月日等欄には「継続」と記載してください。

(3) 主な固定資産等の取得（1件当たりの取引金額が100万円以上の取引を上位5番目まで記載してください。）

資 産 の 種 類 等	取 得 年 月 日 等	取 引 金 額 等 (税込・税抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	・ ・	円		
	・ ・			
	・ ・			
	・ ・			
	・ ・			

4 平成 年中の特殊事情（顕著な増減事項等及びその理由を記載してください。）

消費税の還付申告に関する明細書 (法人用)

課税期間	. . . ~ . . .
------	---------------

所在地	
名 称	

1 還付申告となった主な理由 (該当する事項に○印を付してください。)

輸出等の免税取引の割合が高い	その他 []
設備投資 (高額な固定資産の購入等)	

2 課税売上げ等に係る事項

(1) 主な課税資産の譲渡等 (取引金額が 100 万円以上の取引を上位 10 番目まで記載してください。) 単位: 千円

資 産 の 種 類 等	譲 年 月 日 渡 等	取 引 金 額 等 (税 込 ・ 税 抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	. .			
	. .			
	. .			
	. .			
	. .			
	. .			
	. .			
	. .			
	. .			
	. .			

※ 継続的に課税資産の譲渡等を行っている取引先のものについては、当課税期間分をまとめて記載してください。
 その場合、譲渡年月日等欄に「継続」と記載してください。輸出取引等は(2)に記載してください。

(2) 主な輸出取引等の明細 (取引金額総額の上位 10 番目まで記載してください。) 単位: 千円

取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)	取 引 金 額	主 な 取 引 商 品 等	所 轄 税 関 (支 署) 名

に 輸 利 出 用 取 す 引 る 等	主 な 金融機関	銀 行 金 庫 ・ 組 合 農 協 ・ 漁 協		本店 ・ 支店 出張所 本所 ・ 支所
		預 金	口座番号	
	主 な 通関業者	氏 名 (名 称)		
		住 所 (所 在 地)		

(1 / 2)

3 課税仕入れに係る事項

(1) 仕入金額等の明細

単位：千円

区 分			① 決 算 額 (税込・税抜)	② ①のうち 課税仕入れに ならないもの	(①－②) 課税仕入高
損益科目	商 品 仕 入 高 等	①			
	販売費・一般管理費	②			
	営 業 外 費 用	③			
	そ の 他	④			
	小 計	⑤			
区 分			① 資産の取得価額 (税込・税抜)	② ①のうち 課税仕入れに ならないもの	(①－②) 課税仕入高
資産科目	固 定 資 産	⑥			
	繰 延 資 産	⑦			
	そ の 他	⑧			
	小 計	⑨			
課税仕入れ等の税額の合計額			⑩	⑤＋⑨の金額に対する消費税額	

(2) 主な棚卸資産・原材料等の取得（取引金額が100万円以上の取引を上位5番目まで記載してください。） 単位：千円

資 産 の 種 類 等	取 年 月 日 等 得 等	取 引 金 額 等 (税込・税抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			

※ 継続的に課税資産の取得を行っている取引先のものについては、当課税期間分をまとめて記載してください。
その場合取得年月日等欄に「継続」と記載してください。

(3) 主な固定資産等の取得（1件当たりの取引金額が100万円以上の取引を上位10番目まで記載してください。） 単位：千円

資 産 の 種 類 等	取 年 月 日 等 得 等	取 引 金 額 等 (税込・税抜)	取 引 先 の 氏 名 (名 称)	取 引 先 の 住 所 (所 在 地)
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			
	・ ・			

4 当課税期間中の特殊事情（顕著な増減事項等及びその理由を記載してください。）

(2 / 2)



1. 消費税の改正事項

(3) 平成23年度改正のうち平成24年4月1日以後適用される重要事項

③ 事業者免税点制度の適用要件の追加

- 1) 平成25年1月1日以後開始する年から事業者免税点制度の適用要件が、これまでの基準期間における課税売上高による判定に加えて、**特定期間（前年の1月1日～6月30日の期間）の課税売上高若しくは給与等支払額の合計額（年換算する必要なし）により判定する要件が追加されました。**
- 2) 特定期間の課税売上高により判定するか、給与等支払額の合計額により判定するかは、事業者にとって**有利な判定方法を選択することができます。**
- 3) 給与等支払額の合計額により判定する場合、**従業員・専従者の給与・賞与及びパート・アルバイトの賃金は含まれますが、退職手当又は未払い給与等は含まれず、また、従業員がいない場合や専従者給与のみ場合であっても、判定の対象となります。**
- 4) なお、平成24年1月1日～6月30日の特定期間は、平成25年分課税期間の納税義務の有無の判定に用いられます。

消費税法改正のお知らせ

平成 23 年 9 月
税 務 署

平成 23 年 6 月に消費税法の一部が改正されました。主な改正内容は次のとおりです。

- 改 正 の ポ イ ン ト
- 1 事業者免税点制度の適用要件が見直されました。
当課税期間の前年の 1 月 1 日（法人の場合は前事業年度開始の日）から 6 か月間の課税売上高が 1,000 万円を超えた場合、当課税期間においては課税事業者となります。なお、課税売上高に代えて、給与等支払額の合計額により判定することもできます。
【適用開始時期】平成 25 年 1 月 1 日以後に開始する年又は事業年度から適用されます。
※ 6 か月間の判定期間（「特定期間」といいます。）は平成 24 年 1 月 1 日から始まります。
 - 2 仕入税額控除制度における、いわゆる「95%ルール」の適用要件が見直されました。
当課税期間の課税売上高が 5 億円を超える場合には、個別対応方式又は一括比例配分方式のいずれかの方法により仕入控除税額の計算を行うこととされました。
【適用開始時期】平成 24 年 4 月 1 日以後に開始する課税期間から適用されます。
 - 3 還付申告書への「消費税の還付申告に関する明細書」の添付が義務化されました。
【適用開始時期】平成 24 年 4 月 1 日以後に提出する還付申告書から適用されます。

1 事業者免税点制度の適用要件の見直し

制 度 の 概 要

これまでの事業者免税点制度（法 9①）

①H23. 1. 1～H23. 12. 31	②H24. 1. 1～H24. 12. 31	③H25. 1. 1～H25. 12. 31	④H26. 1. 1～H26. 12. 31
課税売上高 900 万円	課税売上高 2,500 万円	免税事業者	課税事業者

- 基準期間の課税売上高が 1,000 万円を超えると課税事業者となります。

※ 基準期間とは、原則として個人事業者はその年の前々年、法人はその事業年度の前々事業年度をいいます。

これまでの要件に加え、次の要件が追加されました（法 9 の 2）

①H23. 1. 1～H23. 12. 31	②H24. 1. 1～H24. 12. 31	③H25. 1. 1～H25. 12. 31	④H26. 1. 1～H26. 12. 31
課税売上高 900 万円	【特定期間】 1,300 万円 1,200 万円	課税売上高 2,500 万円	課税事業者
		課税事業者	課税事業者

- ②の課税期間の 6 か月間（特定期間）の課税売上高が 1,000 万円を超えると、③の課税期間においては課税事業者となります。

なお、課税売上高に代えて、特定期間の給与等支払額の合計額を用いて判定することもできます。

2 「95%ルール」の適用要件の見直し

制度の概要

一般課税により申告を行う事業者のうち、当課税期間における課税売上割合が95%以上の事業者は、課税売上げに係る消費税額から、課税仕入れ等に係る消費税額の全額を控除することができることとされていましたが、平成24年4月1日以後に開始する課税期間から、当課税期間における課税売上割合が95%以上かつ課税売上高が5億円以下※の場合にのみ全額を控除することができることとされました（法30②）。

したがって、当課税期間における課税売上高が5億円超※の場合、又は課税売上割合が95%未満の場合には、仕入控除税額の計算を個別対応方式若しくは一括比例配分方式のいずれかにより行うこととなります。

※ 当課税期間が1年に満たない場合には、当課税期間の課税売上高を当課税期間の月数で除し、これに12を乗じて算出した金額（年換算した金額）で判定します。

適用開始時期

この改正は、平成24年4月1日以後に開始する課税期間から適用されます。したがって、個人事業者は平成25年分から、事業年度が1年である法人については平成25年3月末決算分から適用されます。

3 「消費税の還付申告に関する明細書」の添付義務化

制度の概要

平成24年4月1日以後、控除不足還付税額のある還付申告書※を提出する場合、「消費税の還付申告に関する明細書」を添付しなければならないこととされました（規22③）。

「消費税の還付申告に関する明細書」は、これまで還付申告書に添付をお願いしていました「仕入控除税額に関する明細書」の記載事項に加え、課税資産の譲渡や輸出取引に係る項目等について記載することとされています。新様式及び記載要領等につきましては、法施行に合わせて国税庁ホームページに掲載していきますので、そちらをご覧ください（法施行時には、税務署の窓口でも入手できます。）。

※ 控除不足還付税額がない申告書（中間納付還付税額のための還付申告書）には添付する必要はありません。

適用開始時期

平成24年4月1日以後に提出する還付申告書から添付する必要があります。

※ 個人事業者の平成23年分及び法人の課税期間の末日が平成24年1月31日となる確定申告については、平成24年3月31日（土）は閉庁日ですから、その申告期限は平成24年4月2日（月）となります。この確定申告について還付申告書を平成24年4月2日に提出する場合には、従来添付していた「仕入控除税額に関する明細書（個人事業者用）/（法人用）」を添付していただいて差し支えありません。

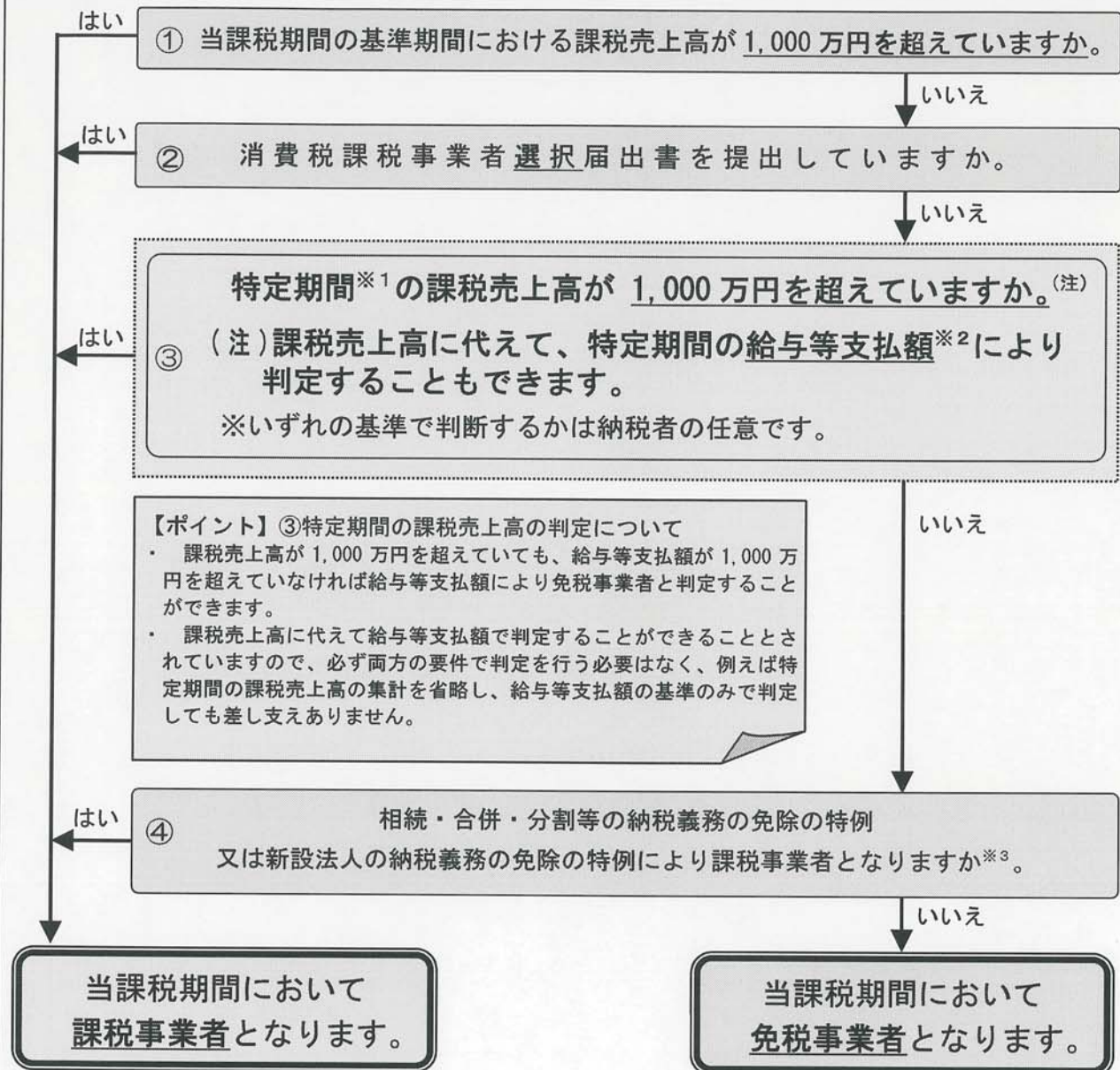
《略語》

法…平成23年度改正後の消費税法

令…平成23年度改正後の消費税法施行令

規…平成23年度改正後の消費税法施行規則

課税事業者判定フローチャート



※1 特定期間とは次の期間をいいます。

個人事業者の場合・・・その年の前年の1月1日から6月30日までの期間

法人の場合・・・原則として、その事業年度の前事業年度開始の日以後6か月の期間

※ 前事業年度が1年でない場合などの特定期間については、上記の特定期間と異なります。詳しくは、次ページの「特定期間の具体的な事例について」をご覧ください。

※2 給与等支払額とは、特定期間中に支払った所得税の課税対象とされる給与、賞与等の合計額です(未払給与等は対象となりません。)。支払明細書の控えや源泉徴収簿から所得税の課税対象とされるものを合計して算出してください。

※3 法第10条から第12条の2第2項までの各規定により課税事業者となる場合をいいます。

※4 特定期間の課税売上高(又は給与等支払額)の判定により課税事業者となる場合は、「消費税課税事業者届出書(特定期間用)」を速やかに所轄の税務署長に提出してください。

適用開始時期

この改正は、平成25年1月1日以後に開始する年又は事業年度から適用されます。

したがって、特定期間(新たな要件による判定期間)は、個人事業者及び事業年度が1年の12月決算法人の場合、平成24年1月1日から6月30日となります。

平成 2 5 年度税制改正大綱

平成 2 5 年 1 月 2 4 日
自 由 民 主 党
公 明 党

(1) 所得税の最高税率の見直し

所得税については、これまでの大幅な累進緩和の結果としてフラット化が進み、わが国経済に格差拡大の傾向が見られる中で、所得再分配機能が低下している。こうした状況を受けて、所得税の最高税率の引上げを行う。その際、平成 26 年 4 月からの消費税率の引上げや、平成 25 年からの復興特別所得税による負担増等にも配慮し、特に高い所得階層に絞って一定の負担増を求めることとし、平成 27 年より、現行の所得税の税率構造に加えて、課税所得 4,000 万円超について 45%の税率を設ける。

(2) 相続税・贈与税の見直し

相続税については、地価が大幅に下落する中においても、バブル期の地価上昇に対応した基礎控除や税率構造の水準が据え置かれてきた結果、課税割合が低下する等、富の再分配機能が低下している。こうした状況を受けて、課税ベースの拡大と税率構造の見直しを行う。

具体的には、平成 27 年より、相続税の基礎控除について、現行の「5,000 万円+1,000 万円×法定相続人数」を「3,000 万円+600 万円×法定相続人数」に引き下げるとともに、最高税率を 55%に引き上げる等、税率構造の見直しを行う。その際、個人の土地所有者の居住や事業の継続に配慮する観点から、小規模宅地等についての相続税の課税価格の計算の特例について、居住用宅地の限度面積を拡大するとともに、居住用宅地と事業用宅地の完全併用を可能とする等の拡充を行う。

また、贈与税の最高税率を相続税に合わせる一方で、高齢者の保有する資産を現役世代により早期に移転させ、その有効活用を通じて「成長と富の創出の好循環」につなげるため、子や孫等が受贈者となる場合の贈与税の税率構造を緩和する等の見直しを行うとともに、相続時精算課税制度について、贈与者の年齢要件を 65 歳以上から 60 歳以上に引き下げ、受贈者に孫を加える拡充を行う。

(3) 消費税引上げに伴う対応

① 住宅取得等に係る措置

住宅投資は内需拡大の柱であり、地域経済への波及効果も大きいものである。住宅取得については取引価格が高額であること等から、平成 26 年 4 月

からの消費税率引上げの前後における駆け込み需要及びその反動等による影響が大きいことを踏まえ、一時の税負担の増加による影響を平準化し、及び緩和するとともに、良質な住宅ストックの形成を促し国民の豊かな住生活を確保するという住宅政策の方向性が損なわれないようにする観点から、住宅ローン減税の拡充をはじめとする税制上の措置を講ずる。

具体的には、所得税において、住宅ローン減税を平成 26 年 1 月 1 日から平成 29 年末まで 4 年間延長することとし、その期間のうち平成 26 年 4 月 1 日から平成 29 年末までに認定住宅（長期優良住宅・低炭素住宅）を取得した場合の最大控除額を 500 万円に、それ以外の住宅を取得した場合には 400 万円にそれぞれ拡充する。このように、消費税率引上げによる住宅需要の減少が最も大きくなると考えられる時期に、特例的な措置として過去最大規模の減税を行う。

更に、自己資金で認定住宅を取得した場合の所得税の住宅投資減税や、住宅リフォームを行った場合の減税措置についても、消費税率引上げに伴う需要を平準化する観点から、拡充する。

平成 30 年以降における住宅にかかる税制のあり方については、上記措置の実施状況を踏まえ、平成 30 年度までの税制改正において検討する。

個人住民税における住宅ローン控除の対象期間を平成 26 年 1 月 1 日から平成 29 年末まで 4 年間延長することとし、その期間のうち平成 26 年 4 月 1 日から平成 29 年末までに住宅を取得した場合の控除限度額を、所得税の課税総所得金額等の 7 %（最高 13.65 万円）に拡充する。

また、この措置による平成 27 年度以降の個人住民税の減収額は、全額国費で補てんする。

所得税に加え個人住民税による住宅ローン減税の拡充措置を講じてもなお効果が限定的な所得層に対しては、別途、良質な住宅ストックの形成を促す住宅政策の観点から適切な給付措置を講じ、税制において当面、特例的な措置を行う平成 29 年末まで一貫して、これら減税措置とあわせ、住宅取得に係る消費税負担増をかなりの程度緩和する。

給付措置の具体的な内容については、税制措置とあわせた全体の財源を踏まえながら検討を進め、一定の周知期間が必要であることを踏まえ、できるだけ早期に、遅くとも今夏にはその姿を示すこととする。

② 車体課税の見直し

自動車取得税及び自動車重量税については、税制抜本改革法第7条第1号カにおいて、国及び地方を通じた関連税制の在り方を見直しを行い、安定的な財源を確保した上で、地方財政にも配慮しつつ、簡素化、負担の軽減、グリーン化を図る観点から、見直しを行うこととされている。

イ 自動車取得税については、安定的な財源を確保して、地方財政への影響に対する適切な補てん措置を講じることを前提に、地方団体の意見を踏まえながら、以下の方向で抜本的な改革を行うこととし、平成26年度税制改正で具体的な結論を得る。

(イ) 自動車取得税は、二段階で引き下げ、消費税10%の時点で廃止する。
消費税8%の段階では、エコカー減税の拡充などグリーン化を強化する。
必要な財源は別途措置する。

(ロ) 消費税10%段階で、自動車税において、自動車取得税のグリーン化機能を踏まえつつ、一層のグリーン化の維持・強化及び安定的な財源確保の観点から、地域の自主性、自立性を高めつつ、環境性能等に応じた課税を実施することとし、他に確保した安定的な財源と合わせて、地方財政へは影響を及ぼさない。

ロ 自動車重量税については、以下の方向で見直しを行うこととし、平成26年度税制改正で具体的な結論を得る。

(イ) エコカー減税制度の基本構造を恒久化する。消費税8%段階では、財源を確保して、一層のグリーン化等の観点から、燃費性能等に応じて軽減する等の措置を講ずる。今後、グリーン化機能の維持・強化及び安定的な財源確保の観点から、環境性能に応じた課税を検討する。

(ロ) 自動車重量税については、車両重量等に応じて課税されており、道路損壊等と密接に関連している。今後、道路等の維持管理・更新や防災・減災等の推進に多額の財源が必要となる中で、原因者負担・受益者負担としての性格を明確化するため、その税収について、道路の維持管理・

更新等のための財源として位置づけ、自動車ユーザーに還元されるものであることを明らかにする方向で見直しを行う。その際、その税収の一部が公害健康被害補償の財源として活用されていることにも留意する。

③ その他消費税引上げに係る措置

イ 軽減税率

- 消費税率の10%引き上げ時に、軽減税率制度を導入することをめざす。
- そのため与党税制協議会で、速やかに下記事項について協議を開始し、本年12月予定の2014年度与党税制改正決定時まで、関係者の理解を得た上で、結論を得るものとする。
- 与党税制協議会に軽減税率制度調査委員会を設置し、適宜、検討状況と与党税制協議会に中間報告をする。
- 協議すべき課題
 - ・ 対象、品目
 - ・ 軽減する消費税率
 - ・ 財源の確保
 - ・ インボイス制度など区分経理のための制度の整備
 - ・ 中小事業者等の事務負担増加、免税事業者が課税選択を余儀なくされる問題への理解
 - ・ その他、軽減税率導入にあたって必要な事項

ロ 転嫁対策

今般の一体改革では、二度にわたる消費税率の引上げを予定していることから、中小事業者から価格転嫁に関する不安の声が多く寄せられている。今回の税率引上げにあたっては、下請法の適用対象となっていない大規模小売店と納入業者の間の取引など流通の分野も含め、力のある事業者による転嫁拒否、実質的な値引き強制等が行われないよう、より踏み込んだ転嫁対策を強力に推進していく必要がある。このような観点から、自由民主党及び公明党は、税制抜本改革法案の審議過程において、円滑かつ適正な転嫁を確保するための独占禁止法及び下請法の特例に関する法制上の措置を講ずるべきと主張し、その旨の規定を追加したところである。この法制

上の措置の具体化を含め、強力な実効性のある転嫁対策を実現する。

3 復興支援のための税制上の対応

東日本大震災からの復興は国民の悲願であり、日本経済を再生させていく上からも、それを目に見える形で前進・加速させることが不可欠である。

高台移転を更に推進するため、一定の要件を満たす防災集団移転促進事業で行われる土地の買取りに係る譲渡所得に対し、5,000万円の特別控除を可能とする制度を創設する。

また、消費税率引上げに伴う負担増により、被災者の方々の住宅再取得等が滞ることのないよう、再建住宅の取得等に係る住宅ローン減税については、被災地以外における住宅ローン減税を上回る拡充を行う。具体的には、平成26年4月1日以降平成29年末までの間に東日本大震災の被災者が新たに再建住宅を取得等する場合、住宅ローン減税の最大控除額を現行の360万円から600万円に引き上げる。

さらに、被災者については、住宅ローン減税の拡充措置に加えて適切な給付措置を講じることにより、復興まちづくりに係る区域指定や宅地造成の時期など外的な要因により被災者間で生じる負担の不均衡を避ける必要がある。このため、住宅の再取得等に係る標準的な消費税の負担増加に対応し得る措置を講じるものとする。給付措置の具体的な内容については、一定の周知期間が必要であることを踏まえ、できるだけ早期に遅くとも今夏にはその姿を示すこととする。

平成30年以降の東日本大震災の被災者の住宅再取得等に伴う負担軽減措置については、集中復興期間（平成27年度まで）における防災集団移転促進事業等の進捗状況その他の事情を勘案し、検討する。

福島については、避難解除区域等への企業誘致を促進するため、避難解除区域等において被災者を雇用する場合の税額控除制度、及び設備投資を行う場合に即時償却や税額控除ができる制度について、新たに避難解除区域等に進出する企業が同様の措置の適用を受けることが出来ることとする。

このほか、東日本大震災に係る津波により甚大な被害を受けた区域のうち、市町村長が指定する区域における土地及び家屋について、固定資産税及び都市計画税を免除する措置等の適用期限を1年延長する。

消費税率に対する取組み例 ～ あるシステムでの事例 ～

2013年 2月22日

SERC Dグループ
研究員 弘中 茂樹

1 / 19 ページ

I アウトライン

1 システム

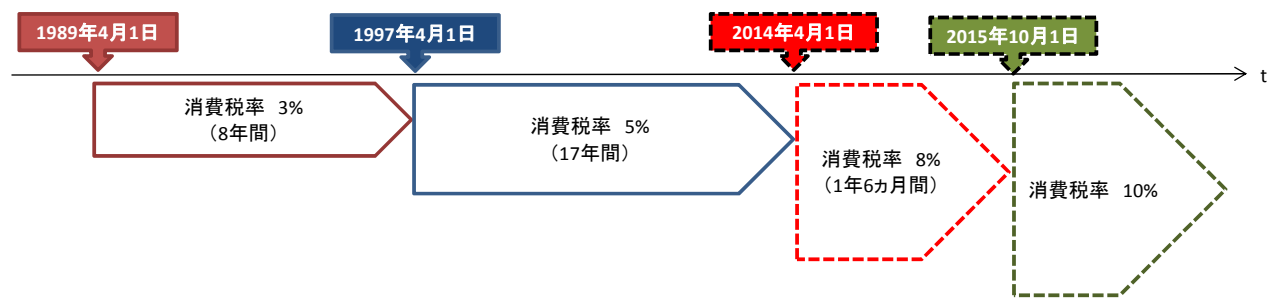
- 資産運用管理のミドル(運用実績入力)～バック(勘定管理)の業務をカバー
- 初期構築から、25年超稼働
 - 基盤層の更改(汎用機→オープン機)
 - ミドル層(DBMS)の更改
 - アプリ層の更改(COBOLコンパイラ変更、一部JAVA化)
 - 消費税率3%、5%変更を対応

業務ロジックの変更はない

2 プロフィール

- 上記システムの初期構築(要件定義)から参画
- 保守開発体制のリーダー、マネージャとして20数年参画

II 消費税率の変遷



III 対応内容

1 消費税率3%導入時

一言で云えば…、 **個別対応(どたばた)**

業務面: 適用品目(科目)、適用タイミングや範囲の確定

		1989年4月1日	
対応種類	内容	施行日以前	施行日以降
個別管理	個々の金額と、消費税額の管理(例 売買金額)	管理しない	管理する
累計管理	個別管理の集約管理(例 B/S)	管理しない	管理する

- 消費税額(切下げ) = 本体価格 * 消費税率 ÷ 100
- 税込金額 = 本体価格 + 消費税額
- 税込金額(切下げ) = 本体価格 * (1 + 消費税率)

システム面: 個別管理、累計管理ロジック追加 ※引落(償却、取消)ロジック追加
予備エリアに項目追加(テーブル)

基本的な業務ロジック

保持情報			求める情報			必要な処理	対応	評価	
#	本体金額	消費税額	税込金額	本体金額	消費税額				税込金額
1	○	—	—	—	○	○	都度、税額計算が必要 消費税額(切捨て) = 本体金額 * 消費税率 ÷ 100 税込金額 = 本体金額 + 消費税額	共通化	基本
2	—	○	—	○	—	○	論外(誤差大)	—	×
3	—	—	○	○	○	—	都度、税額計算が必要 消費税額(切捨て) = 税込金額 * 消費税率 ÷ (100 + 消費税額) 本体金額 = 税込金額 - 消費税額	個々に実装	△
4	○	○	—	—	—	○	都度、加減算が必要		○
5	○	—	○	—	○	—	例) 税込金額 = 本体金額 + 消費税額		
6	—	○	○	○	—	—			
7	○	○	○	—	—	—	追加処理不要	—	◎

計算(取得)した時点で、全項目が保存され、
かついつでも取り出せる状態がベスト

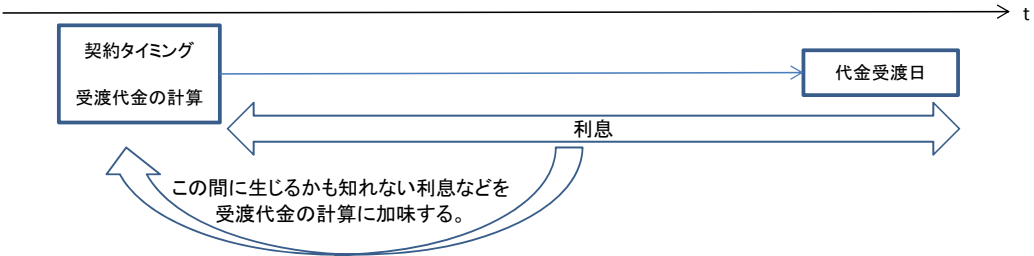
でも、実態は領域不足などで項目の追加が出来ない！ため
① 本体金額のみ ② 税込金額のみ保存のケースも多発

5 / 19 ページ

1) 個別管理

適用品目(科目):どの金額(売買金額、利息、報酬、手数料、…etc.)に対して、適用するの？
→ (結果的には)ほとんどの金額に対して、適用された。
ただし、様々なガイドブックは入手出来たが、なかなか確信が得られず、
個別の科目(金額)毎に適用の要否を確認しながら、対処した。
→ 次回は、軽減税率導入により、**品目レベル**での適用要否、税率の管理が求められる。

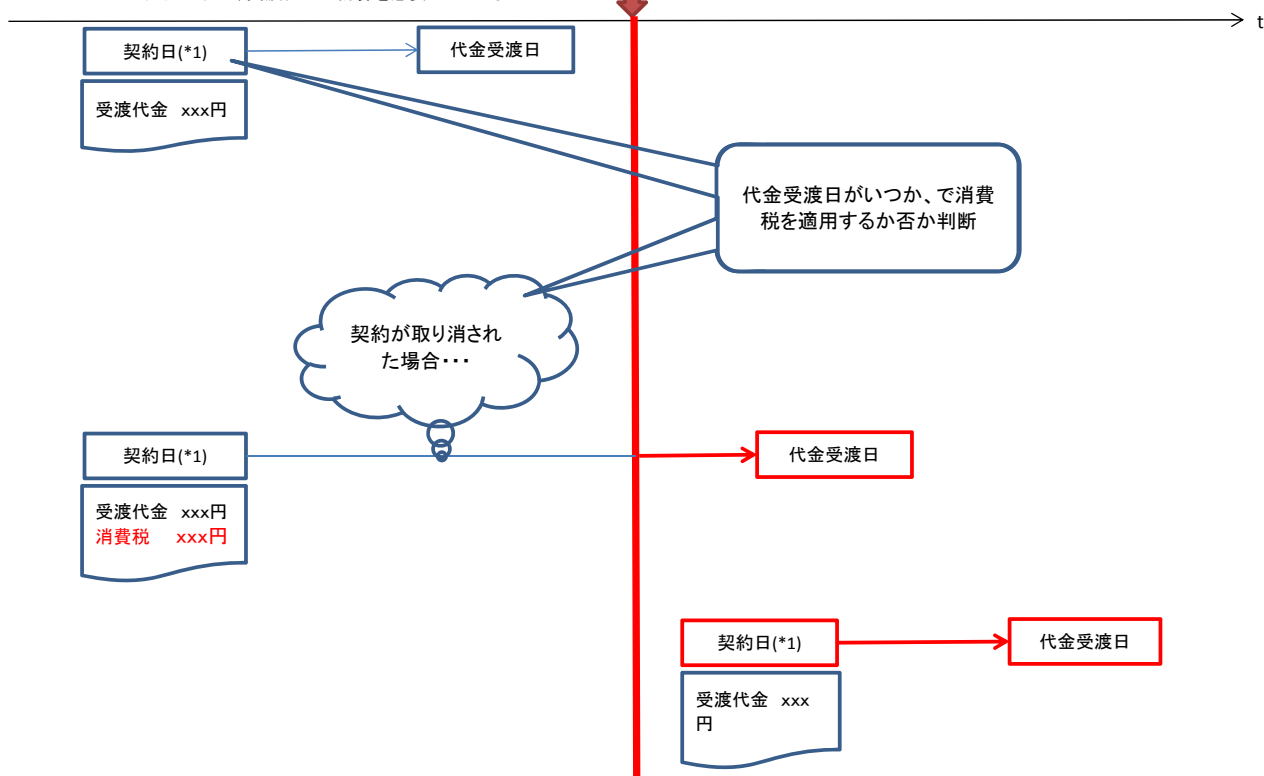
適用タイミングや範囲:どのタイミングの金額に対して、適用するの？
(前提:業務上、一事案に対して、契約タイミングと代金受渡タイミングでの管理があり、
契約タイミングで、代金の計算を行っている)



→ 代金受渡タイミングが、税率適用期間か否かにより、適用要否を決定する。
→ 利息など日々発生する金額も、施行日以降は適用される。

*1:このタイミングで、受渡代金の計算を必要としている

1989年4月1日



7 / 19 ページ

システム面:

外部設計(画面、帳票、テーブル(*1)など)では…、
 税込金額、本体金額、消費税額、消費税率の入出力設計
 → 入出力領域はあるの?(ない場合は)どの項目を対象とする?
 ※ 予備エリアを潰すなど、緊急避難的な対応に終始した。

業務ロジックでは…、

```

if 消費税適用品目 then
  if 消費税適用範囲 then
    消費税額計算、編集、表示
  else
    消費税額ゼロ(処理なし)
  endif
endif
else
  消費税額ゼロ(処理なし)
endif
  
```

システムの観点では、いずれもゼロである。
 業務の観点では、以下の**区別を求められる**事もあった。
 ① 消費税率が適用されなかった"ゼロ" ② 適用した結果の"ゼロ"

支払日: 1989年 3月 31日

受渡金額: 123,456円

支払日: 1989年 4月 1日

受渡金額: 123,456円(3,595

支払日: 1989年 4月 1日

受渡金額: 6円(0

※ ただし、税率は、(新たなI/Oを整備するにはリスクが高かったため)
 税率の定義箇所を限定しながらも、ハードコーディングに近い状態。

2) 累計管理

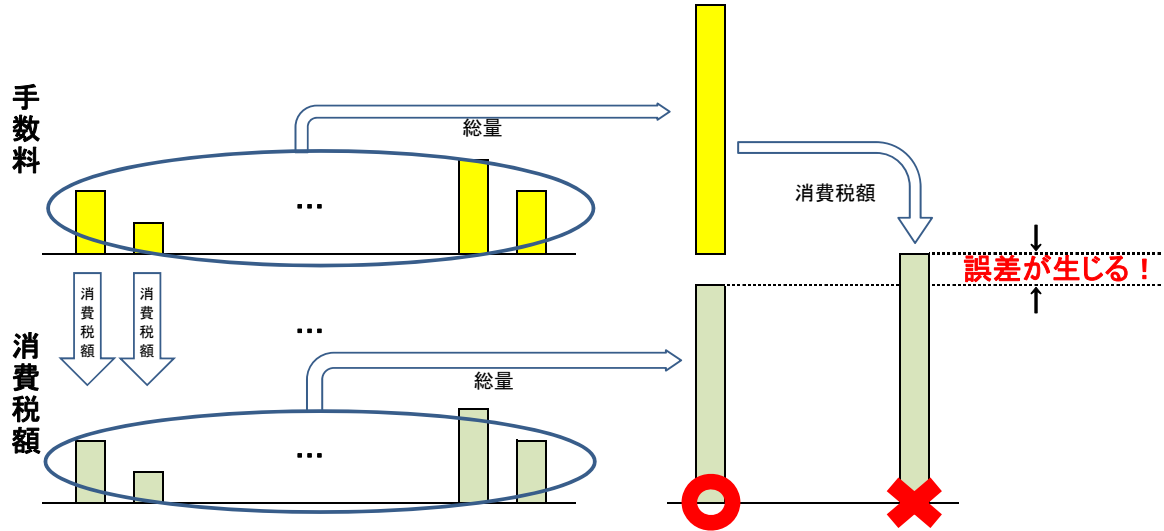
累計管理は、B/Sが代表例であり、基本的には、個別管理の累計である。

→ データ構造により、難易度が異なる。

明細単位で個別管理結果を保有出来ていれば、管理(対応)は容易。

集約(累計値)レベルのみで管理していると…、(施行日の)跨ぎに、複雑な管理が発生する。

例) 日々発生する手数料を管理し、ある時点(決算時など)で支払う(受取る)業務
手数料に消費税が適用された、とすると…どちらの消費税額を採用するの？

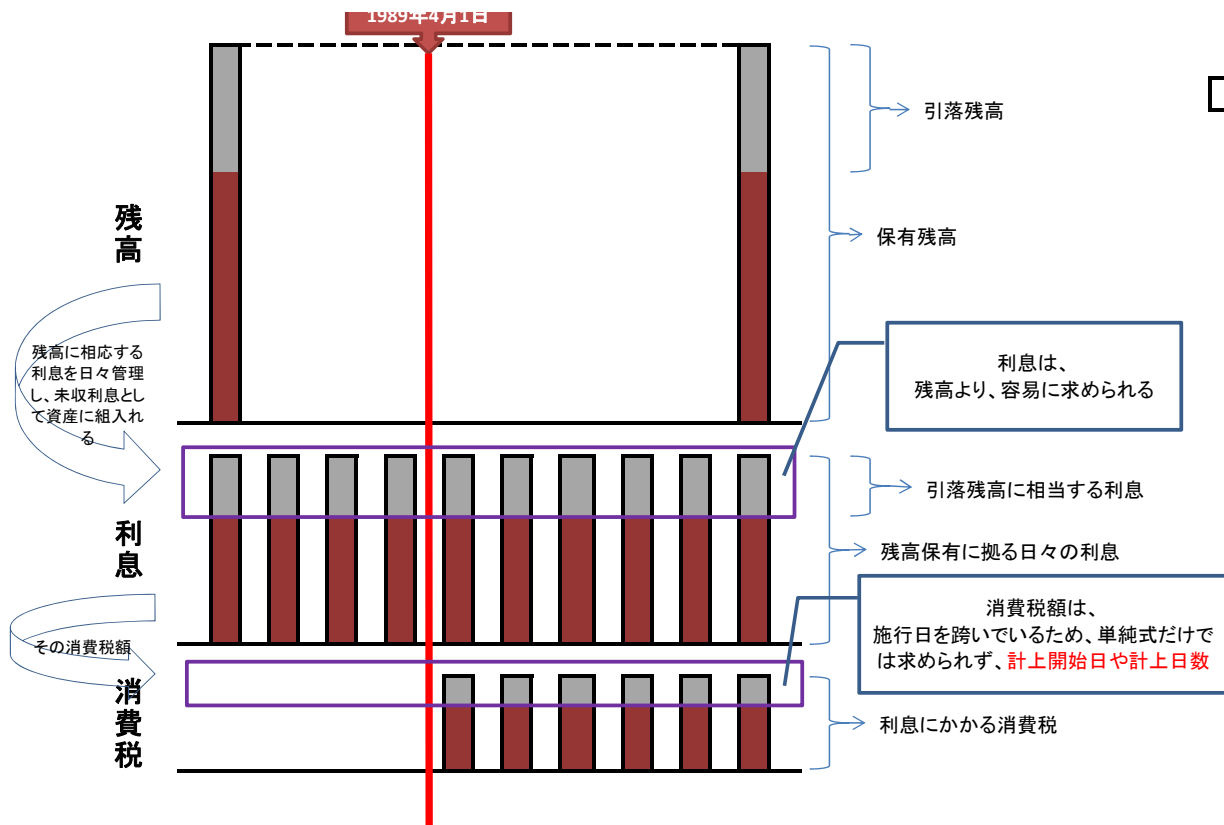


100-49≠51の管理

(減算時、残高値、減算値のいずれの値を採用するかにより、異なる結果が求められる)

1989年4月1日

9 / 19 ページ



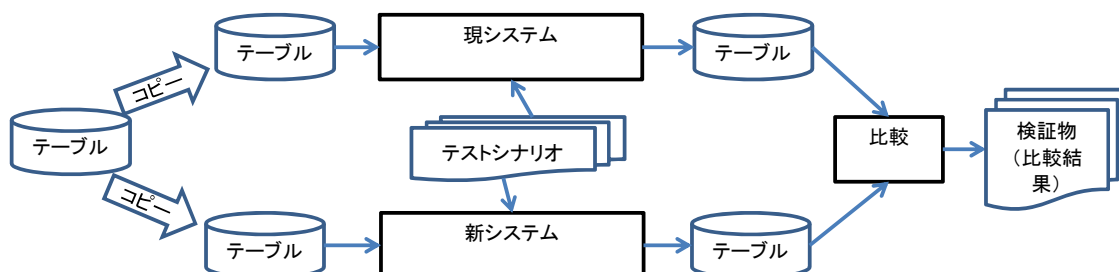
減算値を優先させた場合	残高を優先させた場合
<p>年利1%の商品を、残高36,500千円で10日間運用していた。</p> <p>未収利息累計額＝36,500千円*1%÷365日*10日間＝100円/日*10日間＝1,000円 未収利息消費税累計額＝100円/日*消費税率3%*10日間＝30円</p> <p>残高17,885千円を引落した(売却とか償却とか)場合、 異動残高17,885千円、残高36,500千円－17,885千円＝18,615千円となる。</p> <p>未収利息消費税累計額はどうか？</p>	
<p>異動額を確定させて、残高を求める。</p> <p>異動未収利息累計額＝17,885千円*1%÷365日*10日間＝49円/日*10日間＝490円 異動未収利息消費税累計額＝49円/日*3%*10日間＝10円</p> <p>残高未収利息累計額＝1,000円－490円＝510円 残高未収利息消費税累計額＝30円－10円＝20円</p> <p>※ 残高全額を引落す場合、消費税累計額に端数が残るケースがある！！</p>	<p>残高を確定させて、異動額を求める。</p> <p>残高未収利息累計額＝18,615千円*1%÷365日*10日間＝51円/日*10日間＝510円 残高未収利息消費税累計額＝51円/日*3%*10日間＝10円</p> <p>異動未収利息累計額＝1,000円－510円＝490円 異動未収利息消費税累計額＝30円－10円＝20円</p>
<p>(当然ながら)結果が異なる！</p>	

テストは…

○ テーブルの現新マッチング

現システム、新システムで同じテストシナリオを実施し、
テーブルの現新マッチングを検証の軸とした。
→ テーブルが項目追加のみなので、機械的な現新マッチングが効力を発揮した。

- ※ 消費税関連項目のみ差分が発生し、妥当である事を確認した。
- ※ 画面、帳票などは、単体テストでチェックアウトした。



2 消費税率3%→5%引上げ時

一言で言えば…、 **3%導入時の整備、集約(システム化)**

あちこちに散らばったハードコーディングの共通化、入出力の整備、統一
来るべき(?)消費税率2桁台への準備

業務面: 施行日前後で、対応する消費税率を適用する

→ **旧税(3%)、新税(5%)を、分別管理**することを求められた。

→ 今回も、国税、地方税の比率が異なるため、分別管理が必要か？

#	消費税率	うち国税	うち地方税
1	3%	3%	—
2	5%	4%	1%
3	8%	6.5%	1.5%
4	10%	7.8%	2.2%

1997年4月1日

対応種類	内容	施行日以前	施行日以降
個別管理	個々の金額と、消費税額の管理(例 売買金額)	3%を適用管理する	5%を適用管理する
累計管理	個別管理の集約管理(例 B/S)	3%を適用管理する	5%を適用管理する

システム面: 税込金額、本体金額、消費税額の関係、および入出力を整備
税率、税額の取得処理を共通化

→ データ切替(移行)が発生した。

13 / 19 ページ

1) 個別管理

適用品目、適用タイミングや範囲: 施行日前後で、それぞれの税率を適用する。

システム面: 施行日、税率の定義テーブルを新設(アクセスも共通化)

→ 税率の入出力は、2桁定義で統一

#	期間from(施行日)	期間to	税率
1	00000000	19890331	00
2	19890401	19970331	03
3	19970401	99999999	05
3	19970401	20140331	05
4	20140401	20150930	08
5	20151001	99999999	10

業務ロジックは…、

3%導入時

```
if 消費税適用品目 then
  if 消費税適用範囲 then
    消費税額計算、編集、表示
  else
    消費税額ゼロ(処理なし)
  endif
else
  消費税額ゼロ(処理なし)
endif
```

5%引上げ時(必要ロジック)

```
if 消費税適用品目 then
  if 消費税5%適用範囲 then
    消費税5%額計算、編集、表示
  else
    if 消費税3%適用範囲 then
      消費税3%額計算、編集、表示
    else
      消費税額ゼロ(処理なし)
    endif
  endif
else
  消費税額ゼロ(処理なし)
endif
```

実装は…

共通ルーチン化

税率テーブル

2) 累計管理

累計管理は、個別管理の累計であり、ほぼ3%導入時を踏襲した。

→ 旧税、新税の分別管理が、求められた。

→ (施行日から1年間の限定要請)新たに簡易帳票を提供し、簡便して貰ったケースがある。

勘定系の管理

→ 最長1年で決算を迎え、(消費税額もクリアされる業務であるため)

消費税率引上げの施行日が、1年超の間隔であれば、最大2種の消費税率の管理で充分

→ 8%から10%引上げも、1年6ヵ月の幅があるので、大丈夫

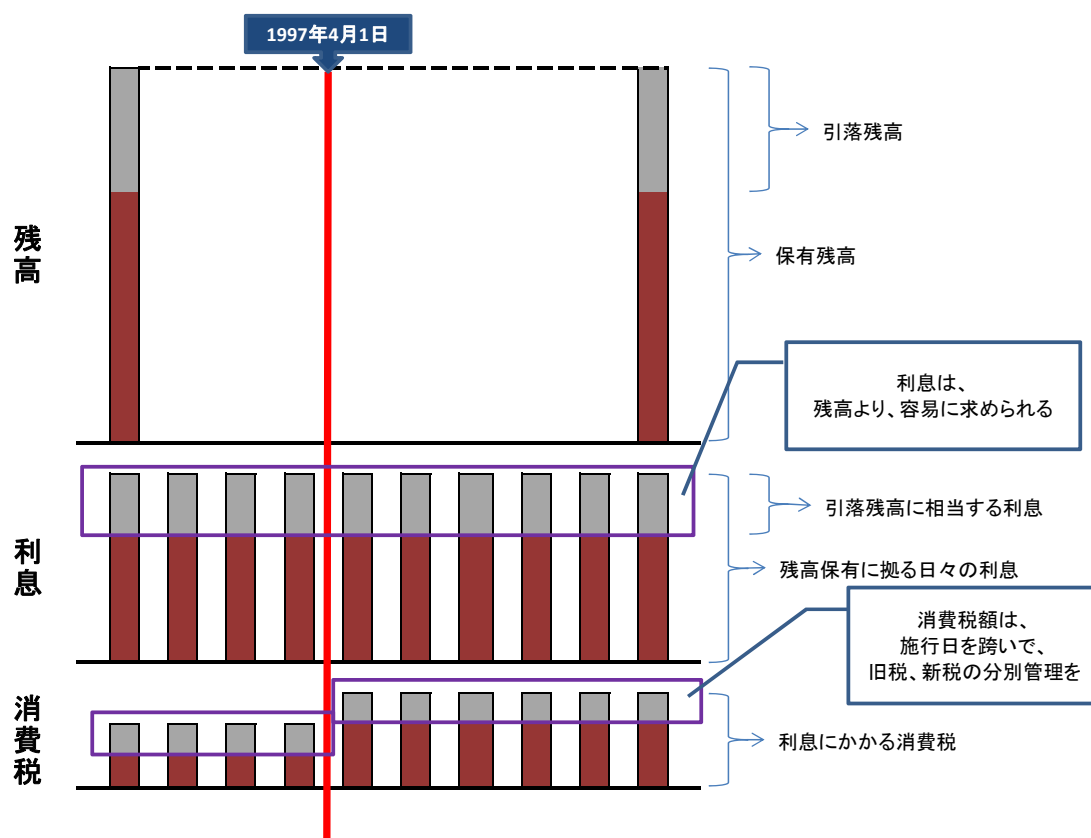
情報系の管理

→ 管理期間3年超なので、3種の消費税率の管理が必要

ただし、勘定系で管理している情報で可能な範囲をカバーし、

情報系独自に、新たな消費税ロジックを組入れる事はしなかった。

15 / 19 ページ



テストは…

○ 帳票現新比較

現システム、新システムで同じテストシナリオを実施し、
(比較する)帳票を定め、**目視による現新比較**を、検証の軸とした。
→ テーブル、帳票などのレイアウトが大幅に変更となり、
機械的な比較ツールの準備は、効率ダウンと判断

#	テストシナリオの観点	現(3%)システム	新(3%→5%)システム
1	施行日以前(消費税率3%)のテストシナリオ	テストシナリオを実施 ↓(帳票取得)	テストシナリオを実施 ↓(帳票取得)
		一致している事を確認	
2	施行日以降(消費税率5%)のテストシナリオ	消費税率のみ3%→5%に変更し て、テストシナリオを実施 ↓(帳票取得)	テストシナリオを実施 ↓(帳票取得)
		一致している事を確認	
3	施行日跨ぎ(累計)の業務	テストシナリオを実施 ↓(帳票取得)	テストシナリオを実施 ↓(帳票取得)
		妥当な相違 がある事を確認	

17 / 19 ページ

今後、こんな事もありそう

■ 面倒だから、1本入れようか…



一合 657円/本 × 10本



○ 徳利1本ずつに消費税が掛った場合、

$$657 * 1.05 * 10 = 6,890円 < 6,570 * 1.05 = 6,898円$$

○ 徳利10本に消費税が掛った場合、

$$657 * 10 * 1.05 = 6,898円$$

(下線は切捨て)



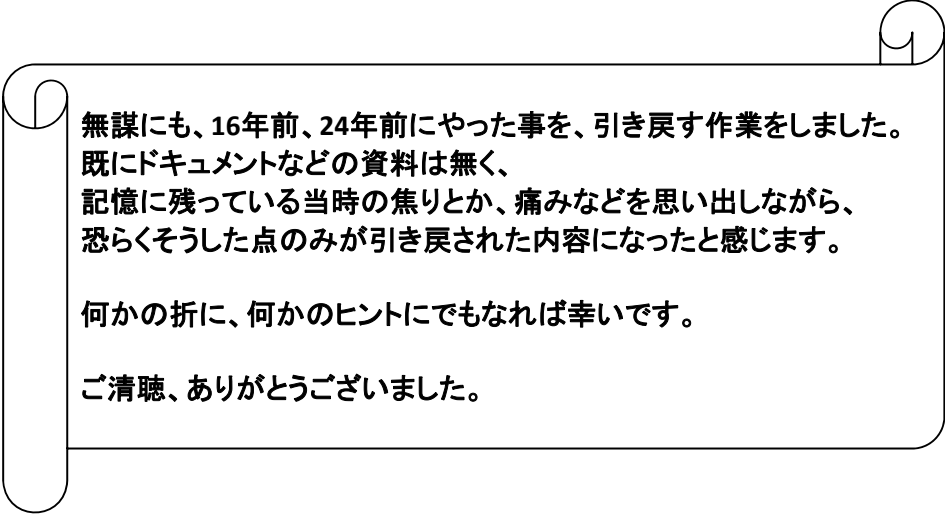
一升 6,570円 × 1本



■ どうしよう…

○ ドーナツ5個以内(外食)→6%、6個以上(食料品)→非課税 【カナダ】

○ ハンバーガーを店内で(外食)→19%、テイクアウト(食料品)→7% 【ドイツ】



無謀にも、16年前、24年前にやった事を、引き戻す作業をしました。
既にドキュメントなどの資料は無く、
記憶に残っている当時の焦りとか、痛みなどを思い出しながら、
恐らくそうした点のみが引き戻された内容になったと感じます。

何かの折に、何かのヒントにでもなれば幸いです。

ご清聴、ありがとうございました。

消費税税率変更に対するシステム変更のポイント

2013年2月22日
ソフトウェア・メンテナンス研究会（SERC）幹事
増井 和也

2013.2.22

©2013 Masui, Kazuya All rights reserved.

1

法律改正対応も立派なソフトの仕事

発表者の自己紹介に代えて

- 貴方は作る人かも？ でも、私は直す人です。
- これまで稼働後直す必要の無い（有価値）ソフトウェアに出会ったことはありません。
- 既存ソフトウェアの修正は多くの人考える以上に面倒な仕事です。
- 消費税率アップに対する必要な既存ソフトウェア修正が後手に回らないよう、早急に既存ソフトウェアの調査開始がされるよう願っています。

2013.2.22

©2013 Masui, Kazuya All rights reserved.

2

本発表の順序

前2氏の講演から、ソフトウェア保守プロセス(次ページ)から見た調査ポイントは何か？

既存ソフトウェアにおける消費税率変更対応(修正・テスト)要否の切分けポイントは何か？

対応(修正・テスト)が必要な場合、どの範囲まで実施する必要があるのか？

‘15年10月消費税率再変更(8%⇒10%)の際、実施が予想される複数税率制導入の備えとして、今から何をしておけばよいか？

影響調査はいつ開始すべきか？

2013.2.22

©2013 Masui, Kazuya All rights reserved.

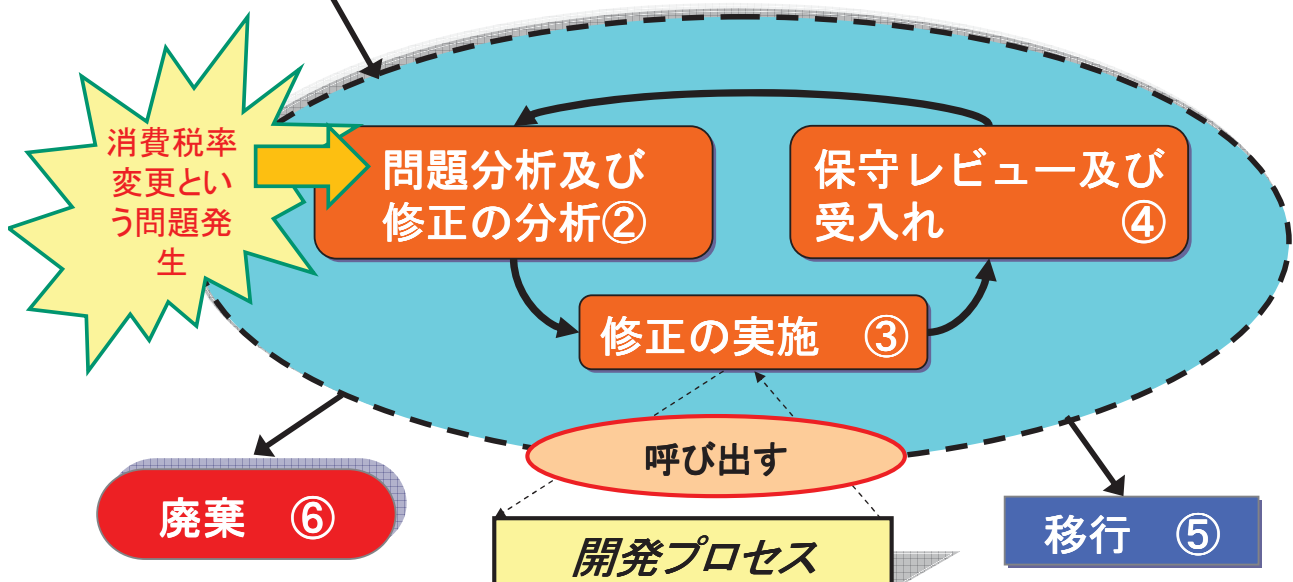
3

ソフトウェア保守プロセスとは

ソフトウェア・メンテナンス研究会編

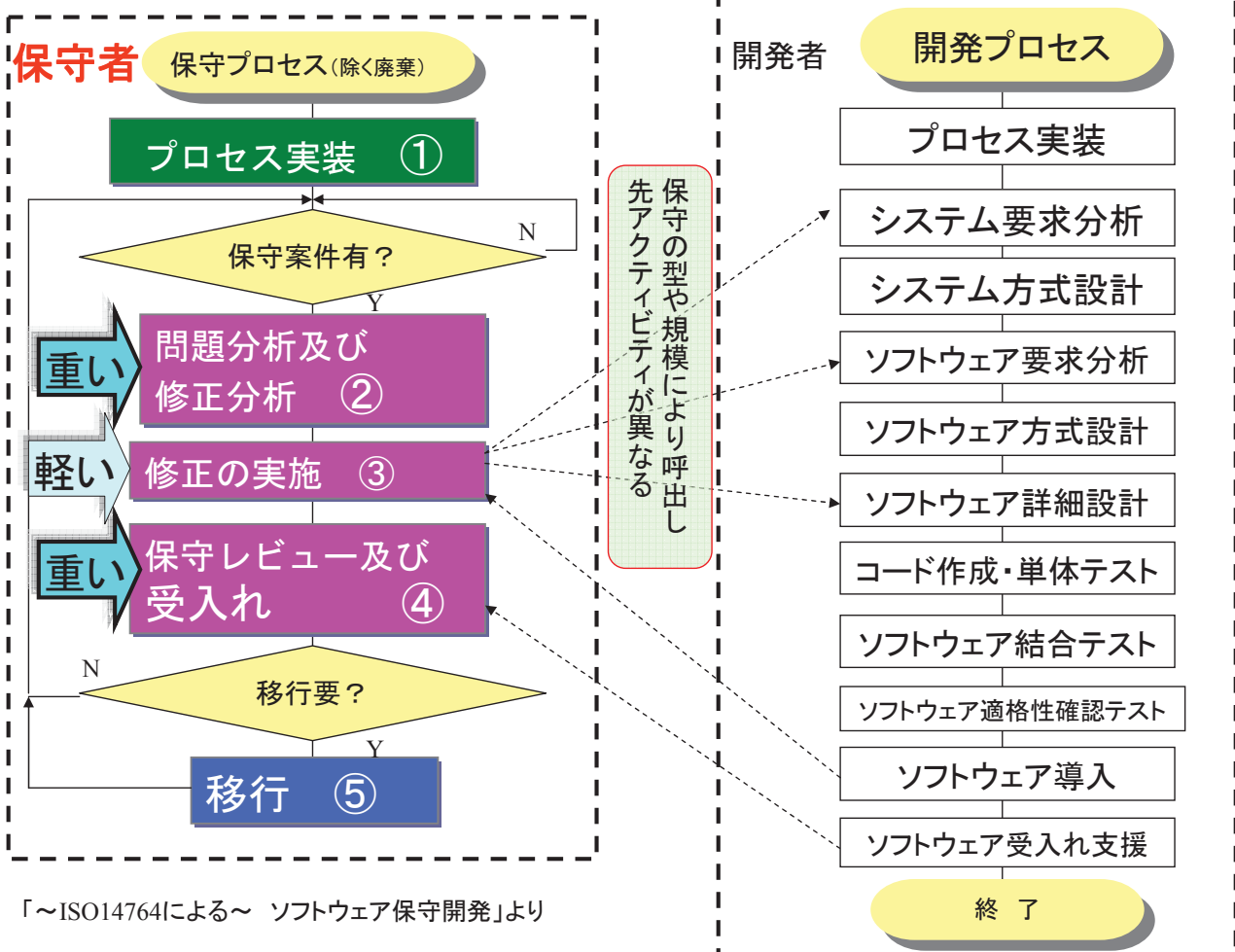
「～ISO14764による～ソフトウェア保守開発」より

プロセス実装 ①



2013.2.22

©2013 Masui, Kazuya
All rights reserved.



1. 消費税率変更で影響受ける処理ポイント

● 次の処理の消費税率が変わるだけ

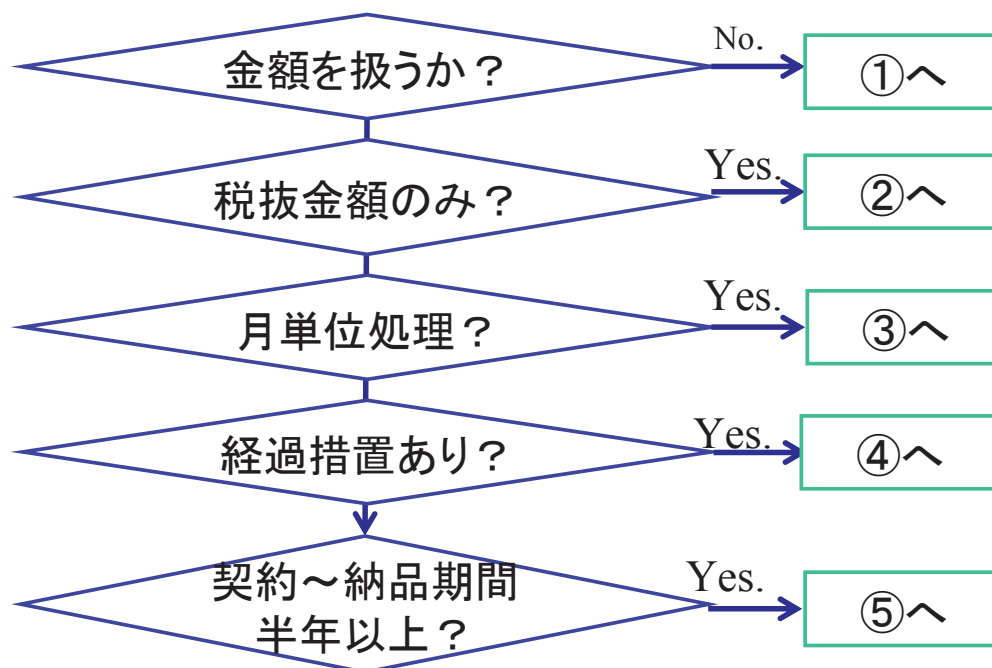
- 税抜金額に消費税を加算する処理
- 税込金額を本体金額と消費税額に分離する処理

● 簡単に済まないキーワード(代表例)

- 旧税率と新税率の混在期間が発生(経過措置期間)
- ('15年10月～)複数税率導入の大きなインパクト
- 既存ソフトウェアの多くが前回('97年4月)以降開発されている(消費税率変更運用を経験していない)

早急な既存ソフトウェアの対応性調査が必要

2. 既存ソフトの対応要否切分けポイント



2013.2.22

©2013 Masui, Kazuya All rights reserved.

7

①金額を処理しないソフトの場合

- 消費税率の基となる金額を扱っていないソフトウェア ⇒ 基本的に消費税率変更の影響を受けない。
- 念のための確認事項
 - 当該ソフトウェアと連携するシステムに金額処理がないか確認(次例)。あれば、当該システムとの関連や連携で影響が発生しないことを確認しておく。

例) 当該システム・・・ユーザの使用量計測システム。課金システムへ使用量(not金額)を送る連携あり。影響例・・・消費税率変更を期に計測精度を見直す変更要求が来る。

2013.2.22

©2013 Masui, Kazuya All rights reserved.

8

②税抜金額のみ処理するソフトの場合

- 消費税を意識しない税抜金額ですべて処理するソフトウェア(生産管理システム等) ⇒基本的に消費税率変更の影響を受けない。
- 念のための確認事項
 - ソースコードに消費税率に関する処理がないことを確認。例:コメントやデータ項目名に「税率」「内税」「外税」「税込」などの記述がないこと。”0.05”や”1.05”や”100/105”といった記述がプログラムに直書きされていないこと。
 - 会計システム等の消費税率を扱うシステムと金額連携していないことを確認。連携している場合は、連携先システムが消費税率変更対応で連携方式に変更がないこと確認。

2013.2.22

©2013 Masui, Kazuya All rights reserved.

9

③月単位で完結するソフトの場合

- 処理が(前月や翌月を含まず)当月で完結、かつ消費税率のマスタ項目が一つ、かつすべての処理が当該マスタ項目を参照するソフトウェア(精算機, レジスタ, 券売機などの組込ソフトウェアも対象) ⇒基本的に当該マスタ項目を消費税率変更するだけでよい。ただし、テストは必要。
- 念のための確認事項
 - 消費税率マスタの整数部は2桁以上か?
 - プログラムに消費税率を意識した直書きの処理はないか?
 - 他システムと税込金額や消費税額を連携する部分があれば、連携先システムが消費税率変更対応で連携方式に変更がないこと確認。
 - 税率変更で顕在化の恐れがある丸め処理の誤差に問題がないか?
 - 過去の帳票出力時、税率マスタ項目を参照する処理がないか?

2013.2.22

©2013 Masui, Kazuya All rights reserved.

10

④税率変更後も旧税率要のソフトの場合

- 会計(納税)・販売管理・購買調達管理・営業支援・固定資産管理・交通費関連などを処理するソフトウェア
⇒消費税率変更後に旧税率が必要となる。消費税率のマスタ項目が1つしかない場合、既存ソフトウェアの大改修が必要となる。
- 必要な確認事項
 - 前回('97年4月)消費税率変更(3%⇒5%)時に対応済みのソフトウェアか？
 - 消費税率のマスタ項目には、新と旧の2項目があるか？
 - 新・旧の消費税率の適用判断をする判断基準がパラメータ化されているか？ ⇒今後の政府の詳細発表への対応性は？

2013.2.22

©2013 Masui, Kazuya All rights reserved.

11

⑤今年10月には対応要のソフトの場合

- 前回('97年4月)と同様な税率変更の経過措置が取られ、かつ今年10月には納期により新税率を適用すべき品目の税込金額を扱うソフトウェア ⇒後7ヶ月で消費税変更対応を行う必要がある。政府から経過措置の扱い詳細発表はギリギリになる可能性あり。
- 必要な確認事項
 - 前回消費税変更(3%⇒5%)時、対応済みシステムか？
 - 前回税率変更時のシステム対応記録(対応計画、テスト計画・仕様書、不具合記録など)が残っているか？
 - 現行システムがどこまで経過措置について考慮されているか？
 - 税率は5%, 8%, 10%の3種類を持てるようになっているか？

2013.2.22

©2012 Masui, Kazuya All rights reserved.

12

3. 対応が必要な場合の影響範囲

- 消費税処理は金額を扱うため、消費税額計算処理変更のみならず、帳票類(※)変更の可能性大。
(※)見積書, 請求書, 注文書, 領収書(レシート), 取引明細書 等(Web公開帳票も含む)
- 既存システムの消費税対応性が不十分である場合, 今後の対応でシステム間連携形式の変更が発生 ⇒ 連携形式が変更となると連携関係にあるシステムで修正やテストが必要となる。
- 修正に対するテスト範囲は, 最終的にシステム全体になることも考えられる(例: 引合⇒見積⇒受注⇒調達⇒検収⇒納品⇒請求⇒回収⇒顧客管理)。

2013.2.22

©2013 Masui, Kazuya All rights reserved.

13

4. 複数税率導入への備え

- 2015年10月に実施の検討がされている複数税率導入について, 今から考える必要があるのか? ⇒ Yes.
- 複数税率導入は, 従来の消費税率変更とは別の概念であり, その考慮がされていない既存システムは再設計が必要となる。
- 何が変わるのか?
 - インボイス方式(次ページ)の導入が考えられる ⇒ 各種帳票をレイアウトから変える必要がある。領収書発行の自動販売機, 精算機の組込ソフトウェアの変更も必要。
 - 今まで総額に対して消費税を計算することも可能であったが, 複数税率が導入されるとすべて品目ごとに消費税を計算・帳票出力する必要がある ⇒ マスタ項目見直しとそれを処理するプログラムロジックの追加・変更・テスト要。

2013.2.22

©2013 Masui, Kazuya All rights reserved.

14

インボイス方式のイメージ

【日本】
請求書

24年4月30日 No.

株式会社〇〇商事 株式会社 △△商事
千代田区霞が関3-1-1

商品名等について包括的な記載可

税込価格のみの記載で可
[適用税率・税額の記載義務はない]

税込合計金額 ¥ 193,500-

品名	数量	単価	金額(税別)	税率	消費税額	税込金額
食料品等	4	1	193,500	10%	19,350	212,850
合計			193,500		19,350	212,850

一定の期間内の取引をまとめて記載可

【イギリス】
A FULL TAX INVOICE

Sales Invoice No.174
O×△ TRADE LTD
From: Any Street, Any Town
To: N.Obody, 222, The High Street, London NE14 4PT

VAT Reg. No.987 6543 21

Sale: Time of supply 16/01/2012 Date of issue: 19/01/2012

Quantity	Description and Price	£ Net of VAT	VAT Rate	Net VAT
6	Radios, SW15 @ £25.20	151.20		
4	Record Players @ £23.6	94.40		
6	Lamps, T77 @ £15.50	93.00	20	67.72
		338.60		
				67.72
				406.32

課税事業者の登録番号

適用税率・税額の記載を義務付け

財務省Webページ『請求書等保存方式』と『インボイス方式』より

15

5. 影響調査はいつから開始すればよいのか？

次のような調査実施阻害要因を乗り越え、早急に調査を開始し、影響有無と対応規模の把握を行うべき。

- 我々のシステムは金額を扱っていないはずなので特になにもなくてよいだろう？
- どの程度影響あるか不明だが、まだ時間があるから(直近の仕事で手一杯なので)もう少し後でもよいのでは？
- 政府から詳細な発表がないので、いまから動いてもしょうがないのでは？
- いざとなれば、17年前の経験者を集めれば何とかできるだろう？
- 当方のシステムはパッケージ(ASP)なのでベンダーが何とかしてくれるだろう？
- 正確に見積もれない状況では対応予算化は難しい。

6. 税率変更の影響調査全般のチェックポイント

品質保証部門やシステム監査とも連携し、楽観的調査ではなく、次のような最悪ケース想定がポイント

- 既存プログラムで設計書や仕様書に明記されていない消費税関連の処理が隠れているかもしれない。
- 税率変更による業務上の影響を十分理解してない担当からの「影響なし」回答は本当に大丈夫か？
- 調査担当からの予想対応工数は十分な品質確認ができるテスト工数も入っているのか？
- 調査は他システムの消費税対応による影響(連携形式変更, システムテスト支援依頼)まで考慮はされているか？
- パッケージ(ASP)ベンダの対応時期や対応版リリース後の必要な受入テスト期間及びテスト体制まで計画されているか？

2013.2.22

©2013 Masui, Kazuya All rights reserved.

17

2. 作業グループ報告書

保守技術者の教育

グループメンバー

アイエックス・ナレッジ(株)

(株) S R A

システム企画研修 (株)

東芝ソリューション (株)

日立ソリューションズ (株)

個人研究員

田中 一夫

古石 ゆみ

上野 則男

沼田 恵助

中山 優紀

塩谷 和範

福島 茂雄

三輪 東

内容

1.	本年度の活動	54
2.	今年度の『教育』というテーマについて	54
3.	テーマの具現化	54
3.1	既存保守ノウハウの教育コンテンツ化	55
3.2	教育コンテンツとして必要なノウハウの補充@三輪	56
4	今年度の教育コンテンツ化を通して気づいたこと	56
4.1	アニメーションはゴールでは無い	56
4.2	コンテンツのリファクタリング	58
5.	次年度の運営とテーマ	60
5.1	運営	60
5.2	テーマ	61
6.	まとめ	61
	付録	63

1. 本年度の活動

2012 年度の活動として、1 回の合宿、5 回のミーティングによる計 6 回の研究会活動を実施した。

表 1.1 1 本年度の活動一覧

No	名称	日程	場所	研究会活動概要
1	研究会キック オフ合宿	2012/11/30- 12/01	葉山	本年度の活動計画の立案
2	第 1 回研究会	2013/01/16	小伝馬町	「保守の教育」のイメージ作り
3	第 2 回研究会	2013/03/06	小伝馬町	「保守の教育」コンテンツ作成 方針決定
4	第 3 回研究会	2013/04/23	品川	コンテンツを整頓するためのタ グ案について方針決定
5	第 4 回研究会	2013/08/23	品川	報告書の方針
6	第 5 回研究会	2013/09/18	品川	報告書

2. 今年度の『教育』というテーマについて

昨年のフォーラムでは試しに映像にしてみた。意外にもその映像への関心が高かった。なぜかをヒアリングしてみると

- ・実際の現場を見ているようで共感を得られる
- ・様々な登場人物の視点・意見を同時に入れられるのが良い
- ・単なるノウハウ集とは違う可能性を感じる

といった意見があがった。つまり、QA のような往復だけの情報ではなく、様々な視点の関係者の意見が交錯する状況がコンテンツとして面白いという意見が非常に多かった。一つの事象に対して様々な関係者を登場させ、多方面の意見を交わさせると面白いのではなかろうかというアイデアから、まず今あるノウハウ集などのコンテンツをビジュアル化して教育コンテンツとして纏めたらどうだろうということになり、今年の活動がスタートした。

期待する効果としては

- ・コンテンツのかたちとして面白く、受け入れられ易い
- ・これまでと違う多様な情報の盛り込みが出来る

といった理由である。

3. テーマの具現化

これまでの成果を新しい教育コンテンツとして纏めていく作業としては次の 2 つを実施した。

(1) 既存保守ノウハウの教育コンテンツ化

(2) 教育コンテンツとして必要なノウハウの補充

3.1 既存保守ノウハウの教育コンテンツ化

教育コンテンツを作る過程の中で重視したポイントは

- ・ 誰向けのコンテンツなのか、誰から誰に対しての教育なのか？
- ・ 見てもらわなければ意味が無い、その為の工夫は？
- ・ コンテンツの検索性をあげて、利用し易くすべきでは？

といったことである。

研究員からは「皆に使ってもらえるものが良い」という目標があり、アイデア出しが行われた。その中で必要なコンテンツはこうだろうと議論するものの、各自の役割や保守担当範囲が異なるからか「何か微妙に違う・・・」といったミスマッチが常につきまとう。しかし「ずれてはいないから、それもありか・・・」といったことが多いまま、発散する状態が続き、結果として、最終的なターゲットを限定することは難しいという判断に落ち着く。ことになる。そうであれば、

(1) それぞれのコンテンツ毎に、対象（誰、何）をターゲットとするのか

(2) 利用する観点からの体系付け

を行うことで結果的に利用者を絞り込めていくであろうと考えた。

また、使い易いという観点にも合致する。

具体的には、動画、ノウハウになっているコンテンツをタグ付けすることで分かり易くしようという考えである。そこで

・ ノウハウの対象（キャリア、分野、工程、種類※後述 3.2）を明確に分類し体系付けるということを行った。

仮に、コンテンツが足りなければ、必要なコンテンツは順次増やせばよい。その繰り返しで網羅性が維持できるのではないかと考えた。

タグの詳細は以下のとおりである

タグ案 A（保守とは何か）

- a) これぞ保守！（でも本当はソフトウェア進化だよ）
- b) 新メンバーが来たよ（半年くらい）
- c) 初心者マネージャーの悩み（半年くらい…3年くらいまで）

タグ案 B（JUAS 情報システム管理の神髄 より）

- a) 変更要求の発生・受付
- b) 影響調査（要件定義なども含む）

- c) 変更実施判定プロセス
- f) 受入 → ベンダーから受け取る
- g) 受入テスト/運用受入テスト (UAT)
- h) 本番移行作業
- i) 後処理 (完了、×、クローズ)

タグ案 C

- a) 心構え/技術

タグ案 D

- a) 製品種別 (受注品/パッケージ品/組み込みなど)

3.2 教育コンテンツとして必要なノウハウの補充@三輪

既存のコンテンツで網羅できていない分野の中で、以下のコンテンツを追加しようという動きも同時に起きた。

- (1) 保守ってそもそも何なのか (「心構え」系)
- (2) 特に保守にこそ求められる技術

これらについては、ノウハウの種類として追加した。

また、それと同時に、保守者ならではの悩み、解決や行き場のない思いも、トピックとして数多く挙がってきた。これらはノウハウではないが、保守者の自覚と問題意識の向上の観点から有用であろうと、今回のノウハウに加えられることにした。(成果物については付録参照)

4 今年の教育コンテンツ化を通して気づいたこと

4.1 アニメーションはゴールでは無い

今期の活動で今後のヒントになる大事な要素は以下です

- ・アニメーションの「実際の現場を見ているようで親近感を持てる」
- ・議論の中の「その考え方は合っているようで何か微妙に違う・・・」といったミスマッチ。しかし「ずれてはいないから、それもありか・・・」
- ・保守者ならではの悩み、解決や行き場のない思いも、トピックとして数多く挙がってきた。
- ・アニメーションはゴールではない、課題の洗い出しが目的ではなかったか？
- ・コンテンツに答えは要らないのでは？ あくまでも一つの例の提示にしか

らないでしょ？ だって現場ごとに違うんだから。今期は始める際には、できるかもしれないなあという期待を込めていたけど、やってみると発散して終わっちゃう。みんな見ている方向が違うから、まとまらないのは当然かなあと、最近思うようになった

上記を言い換えれば

- ・アニメーションは実際の現場のようだ → 事象を正しく捉えている
- ・議論の中で「ずれていない」、でもマッチしない → 同じものと違うものが混在している
- ・問題点、課題、悩み → 結構似ている、共感できる
- ・アニメーションを作るのはゴールではない → 導入のはず
- ・入口は一緒、でも出口はバラバラになる

ということのような気がしています。これは

現場、役割・担当は違うが、問題提起、導入部は意外と同じ場合が多い、だから共感を出来る。つまり、問題の切り口は正しい。しかし、解決策は現場ごとに異なるから、解には微妙な違和感が残る。

そのため、具体的な作業への落とし込みの際、発散しやすいのだと思います。

昨年のレポートでいうところの、

相談	○
問題	○
原因	△
解決策	△

ということになると思います。

△については、決して間違っているわけではありませんが、現場によってはマッチしない場合があり、結果的に「その考え方はおかしい」のような流れに発展しやすく、受け入れられ難いことを意味しています。

つまり、相談、問題は共感できるが、その原因と解決策は現場ごとに異なるので、意見が分かれやすい

といったことが、教育コンテンツを纏めていくことの難しさであろうと思いました。

4.2 コンテンツのリファクタリング

では、結局、教育コンテンツをそろえることは無理なのか？ ですが、次のようにすることで、もっと使い易く、納得感のあるコンテンツ（ノウハウ）を提供できるようなのでは？

保守を語ると、課題・問題点については盛り上がるもの、解決策になると発散する点を踏まえ、思い切ってコンテンツを分けてみると良いと思いました。解決策まで同じコンテンツで提供すると、受け入れ難くなるのだとおもいます。そうなってしまっは、我々の望む、継承・教育・共有・再利用（2. 今年度のテーマにありますね）といった過程まで発展しにくいと考えたからです。

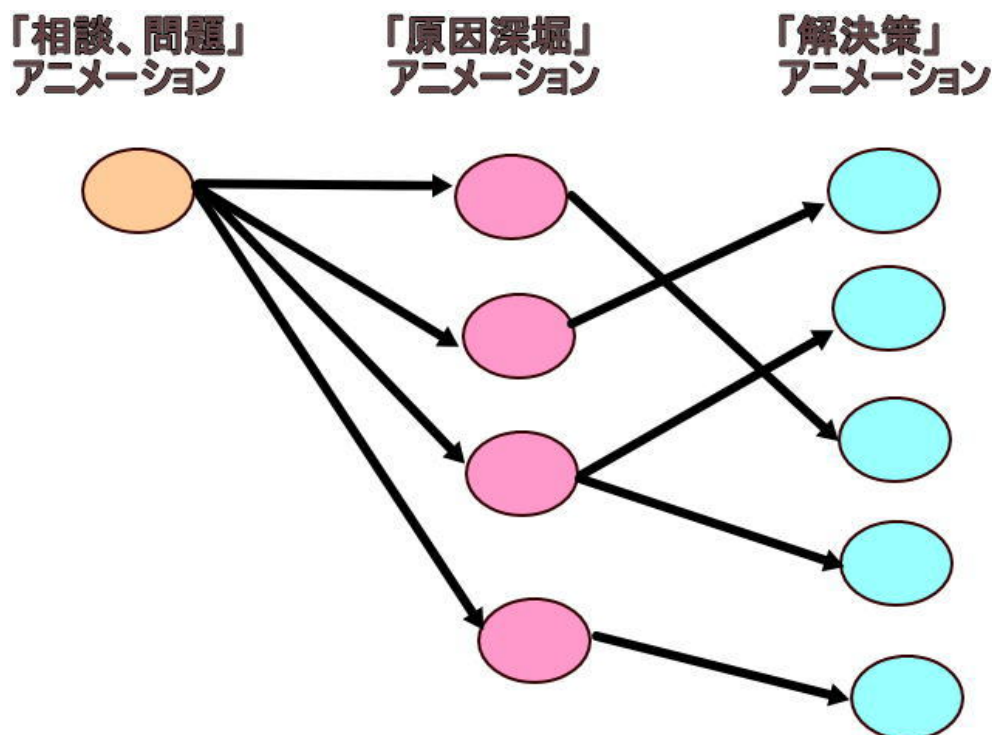
既存のノウハウ集を

「相談、問題」 アニメーション

「原因深堀」 アニメーション

「解決策」 アニメーション以外もあり

に分けて、整理していく。



「相談、問題」コンテンツに、営業、PM、SE 等の役割の視点も盛り込む。

「相談、問題」では、役割や立場の異なる登場人物にあれよ、これよと語らせませす。アニメーションで纏め上げる利点は、発散しているように見えて、まとまりが出る点だと思います。また、ここで自身の立場と観点の違う問題点が提起されたとしても「ああ～、なるほどねえ。」で自分以外のこともすんなり入り込めるところも利点です。

「原因深掘」コンテンツは、いわゆる王道の方向へ収束する流れをつくるものですが、ここは環境によって異なると思いますので、いくつかのコンテンツを用意して、人によってそのコンテンツを選べると良いと思いました
例)

A さん 「相談、問題コンテンツ 1」→「原因深掘コンテンツ 1」

B さん 「相談、問題コンテンツ 1」→「原因深掘コンテンツ 3」

C さん 「相談、問題コンテンツ 1」→「原因深掘コンテンツ 4」

ここでタグ付けが活きてこないかなあと思っています。

原因については、チョイスしてもらいたいイメージです。

チョイスするにしても何も無いと大変ですから、タグ付けが活けるといいなあと思っています。

「解決策」コンテンツは、様々なノウハウ集のイメージです。

今後、様々な解決策やノウハウが充実していくといいなと思っています。

こういったネタをそろえつつ、保守といったものを様々な視点で捉えたコンテンツが揃った、良い情報提供サイト（教育・継承）になるのではないのでしょうか。

対象は、エンジニアだけでなく、広く関係者を取り込み、営業やマネージャーもターゲットにして登場人物の役割に取り入れていくべきでしょう。私はエンジニアだけで保守を語るのは無理と思っています。

例えば

ソフト保守こそビジネス支援のエースの気概を持て

ですが、これは組織の問題が大きく影響しており、エンジニア単体での実現は

無理だと思っています。

これが出来ない現場は、おそらく保守の組織が
認められにくい、誉められにくい、注目されにくい
といった問題を抱えていると思います。

上司は理解のある上司で尊敬できる、でも会社は保守に対して利益の出ない
無駄なビジネスだと思って支援をしてくれない

そんな環境では、気概というより閉塞感に陥りやすいと思います。

ですから保守の活性化はエンジニアだけでなくマネージャーにも示唆を与えていくべきものでしょう。おそらく 10 年後の保守のやり方は大きく変わっていて、重要度も増し、経営の視点がもっともって入ってきます。ですから、そういう視点も入れていくべきだと思いました。

この心構え系の解決策には、マネジメント向けのカテゴリが必要かもしれませんね。エンジニアには興味が無いような気がします。

このように考えると、「相談、問題」-「原因深堀」-「解決策」を纏めてしまうのではなく、いくつかのパターンに派生して提供できるようになると良いと思います。

5. 次年度の運営とテーマ

5.1 運営

A グループでは今年度はリーダーを置かない形で、リーダーの負担が増えることの回避と、負担の分散を期待して運営を行った

もともと、各研究員が業務多忙のため、Face to Face で一つの研究を続けていくのが困難になってきたこともあったため、ここ 2 年くらいは方法を模索している状況であった。しかしその結果、忙しさで優先度が低くならざるを得ず、進捗ははかばかしくなかった。しかしながら、活動は業務とともにあるものであり、方法やペースが変わっていくのは仕方ないことではないかと考えた。

研究員からは、「活動はどうしても個人に頼ってしまいがちです。その人が忙しいとグループ自体が停滞してしまう。複数の活動が、それぞれ主体者を持って走っていれば、一人が忙しくなってもグループとしては動いていられると

思います。」という意見もあり、それが一番实际的であり、今後目指すべき方向だと考えている。

5.2 テーマ

今年度から始めた、教育というテーマについて、継続していきたい。その中で、アイデアとしてあるものは以下のようなテーマである。

(1) 保守に必要な教育(知識・技術)とは？

今年度の議論の中で出てきた「保守の教育」に関する疑問について、継続して議論していきたい。

- ・「保守のための教育」というが、新規開発に必要なそれとの違いは何か。

- ・「保守」であるということ、新規開発との差を認識する必要や、保守のほうがより求められる技術分野があるのではないか。

(2) 保守に関する技法

今後の展望として、「保守現場の課題を解決する」技法についても今後解決策としての技法の追加をして行きたい。

その他にも、各研究員が考える教育コンテンツの拡充を随時企画して行きたいと考えている。

6. まとめ

毎年のことだが、今年度は特に研究員の業務が多忙で開催見送りになった研究会も多かった。

相対的に個別の研究活動が増えたせいか、研究員間での保守に関する問題意識が多岐にわたることがあらわになってきた。

(執筆担当の経験でも、SEA 主催のソフトウェアシンポジウム 2013 で開催した保守ワーキンググループの「保守への帰還」セッションに参加した時にも、参加者は全員がメンテナンス研究会研究員であったが、問題意識が違うことが明確で、問題点や課題の整理はできたが統一見解を出すには至らなかった。)

もともと、保守作業は新規開発作業を含み、かつ対象業務分野及び業務環境の多様さへの対応、新旧ハードウェア/ソフトウェア技術の混在への対処、シ

システム移行への対処、トラブル対応、過少見積もりになりがちな作業環境・期間・人的資源への対処、さらには、短期間でのトラブル原因究明が求められることによる経験豊富な技術者への依存などなど、さまざまな問題がある。

このように、担当業務や部署により優先順位や問題意識は違う。Aグループでは、これまでの意見交換で重要な知見の整理と蓄積をしてきたが、これを共通の保守ノウハウとするのはなかなか難しい。ましてや保守作業の重要性を訴えるためには工夫が必要だとの合意のもとに、今年度はアニメーションによる保守テーマごとに整理した「小芝居」の充実を行った。

このアニメーションによる「小芝居」を、他団体の会合で紹介した際の意見として、保守の重要性を意識させるための良い方法との評価を得たが、さらにどうすれば良いかの示唆を加えてほしいとの意見があったとの報告があり、今後はタグ付けによる分類と合わせて、立場ごとの示唆について検討することにした。

小芝居なのですから、多くの関係者を登場させ、その役割や立場に応じた発言をさせるのがよいと思います。また、小芝居鑑賞の手引（共通）として、関係者のプロフィールを説明（語らせる）する背景資料（アニメーション？）などがあればわかりやすいと思います。

一方個人研究として、現場の技術力を見定め、それを自覚させることを通じての現場主導の技術力向上を図るために、コードインスペクションコンテストの試行報告と提案があった。個々の技術力を高めることは当初からの重要課題であるため、教育テーマの一環として実施方法を検討するなど、今後も取り上げていくことになった。

先にも述べたが、立場や分野などにより保守作業に対する観点は違うことを改めて意識し、整理するなど、多様な視点や示唆などを取り込むことにより、ノウハウ集や小芝居の充実および、保守技術教育に活かしていきたいと考えている。

付録

・成果物一覧表
コンテンツ一覧

以下のタイトルをクリックすると映像
が表示されます(IE実証済)

工程	Know How No	映像コンテンツタイトル
見積もり・計画工程	1.3.3.	過小見積もりのつもりで答えたら、概算見積もりにされてしまいました…(Fさん(36歳)、チームリーダー)
	1.3.4.	少なすぎる予算で、多すぎる要求内容を受託させられました…(Sさん(34歳)、開発担当)
要件定義工程	1.4.1.	要求が曖昧で先に進んでいいか判りません…(Oさん(39歳)、プロジェクトリーダー補佐)
	1.4.2.	修正内容を具体化したところで顧客殿が当初の要件を変更してきます…(Fさん(36歳)、チームリーダー)
	1.4.3.	顧客殿業務の知識不足で、効果的な設計や提案ができません…(Oさん(39歳)、プロジェクトリーダー補佐)
設計工程	1.5.1.	影響範囲分析が楽に出来るドキュメントを教えてください。(Yさん(49歳)、夢見るプロジェクトリーダー)
本番環境での作業工程	1.6.4.	本番データをテストアクセスしそうになりました…(Gさん(24歳)、SE)
引継ぎ	1.8.1.	リーダー交代の引き継ぎが難航します…期間や時間に問題がある場合、どうすればよいのでしょうか？(Kさん(42歳・プロジェクトリーダー))
	1.8.2.	リーダー交代の引き継ぎが難航します…これって前任者と後任者のスキルの差なのかなぁ(Kさん(42歳・プロジェクトリーダー))
	1.8.3.	リーダー交代の引き継ぎが難航します…これって業務の範囲、作業量が問題なのだろうか(Kさん(42歳・プロジェクトリーダー))
	1.8.4.	リーダー交代の引き継ぎが難航します…これって引継ぐべき情報が問題なのだろうか(Kさん(42歳・プロジェクトリーダー))
2. 影響調査（要件定義なども含む）	1.3.3.	過小見積もりのつもりで答えたら、概算見積もりにされてしまいました…(Fさん(36歳)、チームリーダー)
	1.3.4.	少なすぎる予算で、多すぎる要求内容を受託させられました…(Sさん(34歳)、開発担当)
3. 変更実施判定	1.4.2.	修正内容を具体化したところで顧客殿が当初の要件を変更してきます…(Fさん(36歳)、チームリーダー)
2. 影響調査（要件定義なども含む）	1.5.1.	影響範囲分析が楽に出来るドキュメントを教えてください。(Yさん(49歳)、夢見るプロジェクトリーダー)
初心者マネージャーの悩み（半年くらい…3年くらいまで）	1.3.3.	過小見積もりのつもりで答えたら、概算見積もりにされてしまいました…(Fさん(36歳)、チームリーダー)

タグ	Know How No	映像コンテンツタイトル
初心者マネージャー の悩み（半年くらい …3年くらいまで）	1.4.1.	要求が曖昧で先に進んでいいか判りません…(O さん(39 歳)、プロジェクトリーダー補佐)
	1.4.2.	修正内容を具体化したところで顧客殿が当初の要件を変更してきます…(F さん(36 歳)、チームリーダー)
	1.4.3.	顧客殿業務の知識不足で、効果的な設計や提案ができません…(O さん(39 歳)、プロジェクトリーダー補佐)
	1.5.1.	影響範囲分析が楽に出来るドキュメントを教えてください。(Y さん(49 歳)、夢見るプロジェクトリーダー)
	1.8.2.	リーダー交代の引き継ぎが難航します…これって前任者と後任者のスキルの差なのかなぁ(K さん(42 歳・プロジェクトリーダー))
	1.8.3.	リーダー交代の引き継ぎが難航します…これって業務の範囲、作業量が問題なのだろうか(K さん(42 歳・プロジェクトリーダー))
	1.8.4.	リーダー交代の引き継ぎが難航します…これって引継ぐべき情報が問題なのだろうか(K さん(42 歳・プロジェクトリーダー))
これぞ保守！（でも 本当はソフトウェア 進化だよ）/新メン バーが来たよ（半年 くらい）	2.1.1.	SERC A グループ活動計画 2012
	2.1.2	居酒屋プレゼン
	2.1.3	仕様を知っているということ 6 人
	2.1.4	保守って何なの

Know How No			既存ノウハウの工程	動画 番号	タグ案A	タグ案B	備考
1.3.1.	経験により見積もり能力に差が出るのですが…(Yさん(49歳)、プロジェクトリーダー)	3	見積もり・計画工程		初心者マネージャーの悩み(半年くらい…3年くらいまで)	2. 影響調査(要件定義なども含む)	
1.3.2.	短い時間で行う“当初見積もり”の精度を高めたいのですが…(Yさん(49歳)、プロジェクトリーダー)	4	見積もり・計画工程		初心者マネージャーの悩み(半年くらい…3年くらいまで)	2. 影響調査(要件定義なども含む)	
1.3.3.	過小見積もりのつもりで答えたら、概算見積もりにされてしまいました…(Fさん(36歳)、チーム	5	見積もり・計画工程	1	初心者マネージャーの悩み(半年くらい…3年くらいまで)	2. 影響調査(要件定義なども含む)	
1.3.4.	少なすぎる予算で、多すぎる要求内容を受託させられました…(Sさん(34歳)、開発担当)	7	見積もり・計画工程	2		2. 影響調査(要件定義なども含む)	
1.3.5.	直感に頼るのではなく根拠に基づく変更規模の見積もりをしたい…(Oさん(39歳)、プロジェクトリーダー補佐)	9	見積もり・計画工程		初心者マネージャーの悩み(半年くらい…3年くらいまで)	2. 影響調査(要件定義なども含む)	
1.4.1.	要求が曖昧で先に進んでいいか判りません…(Oさん(39歳)、プロジェクトリーダー補佐)	10	要件定義工程	3	初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.4.2.	修正内容を具体化したところで顧客殿が当初の要件を変更してきます…(Fさん(36歳)、チームリーダー)	13	要件定義工程	4	初心者マネージャーの悩み(半年くらい…3年くらいまで)	3. 変更実施判定	
1.4.3.	顧客殿業務の知識不足で、効果的な設計や提案ができません…(Oさん(39歳)、プロジェクトリーダー補佐)	15	要件定義工程	10	初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.5.1.	影響範囲分析が楽に出来るドキュメントを教えてください。(Yさん(49歳)、夢見るプロジェクト	18	設計工程	11	初心者マネージャーの悩み(半年くらい…3年くらいまで)	2. 影響調査(要件定義なども含む)	
1.5.2.	修正方法と意図が正しく伝わらなくて…(Nさん(32歳)、チームサブリーダー)	20	設計工程		初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.6.1.	リリース手順がないシステムでのリプレースを実施することになりました…(Sさん(34歳)、開発担	21	本番環境での作業工程			8. 本番移行作業	
1.6.2.	リリース担当者が、退職し引き継ぎも無い状態で、リリースを担当する事になりました…(Sさん(34歳)、開発・運用担当)	22	本番環境での作業工程			8. 本番移行作業	
1.6.3.	ハードウェアの保守切れに伴い、ミドルウェアも入れ替える必要が出てきたが、ミドルウェアの資料が無いことに気付いた…(Sさん(34歳)、開	23	本番環境での作業工程			2. 影響調査(要件定義なども含む)	
1.6.4.	本番データをテストアクセスしそうになりました…(Gさん(24歳)、SE)	24	本番環境での作業工程	5			
1.7.1.	不具合を恒久対応するのに必要な工数を交渉したいのですが…(Kさん(42歳・プロジェクトリー	26	不具合対応		初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.7.2.	不具合対応の工数確保に苦労しています…(Gさん(24歳)、SE)	27	不具合対応				
1.8.1.	リーダー交代の引き継ぎが難航します…期間や時間に問題がある場合、どうすればよいのでしょうか？(Kさん(42歳・プロジェクトリーダー))	28	引継ぎ	12			
1.8.2.	リーダー交代の引き継ぎが難航します…これって前任者と後任者のスキルの差なのかなあ(Kさん(42歳・プロジェクトリーダー))	32	引継ぎ	6	初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.8.3.	リーダー交代の引き継ぎが難航します…これって業務の範囲、作業量が問題なのだろうか(Kさん(42歳・プロジェクトリーダー))	34	引継ぎ	13	初心者マネージャーの悩み(半年くらい…3年くらいまで)		
1.8.4.	リーダー交代の引き継ぎが難航します…これって引継ぐべき情報が問題なのだろうか(Kさん(42歳・プロジェクトリーダー))	35	引継ぎ	14	初心者マネージャーの悩み(半年くらい…3年くらいまで)		
	SERC Aグループ活動計画2012			7	これぞ保守！(でも本当はソフトウェア進化だよ)/新メンバーが来たよ(半年くらい)		
	居酒屋プレゼン			8	これぞ保守！(でも本当はソフトウェア進化だよ)/新メンバーが来たよ(半年くらい)		
	仕様を知っているということ6人			9	これぞ保守！(でも本当はソフトウェア進化だよ)/新メンバーが来たよ(半年くらい)		

	保守って何なの			15	これぞ保守！（でも本当はソフトウェア進化だよ）/新メンバーが来たよ（半年くらい）		
	「システムは稼動してからが本番」と経営トップが認識し、保守・運用工程に対してしっかり投資						システム保守・運用で失敗しない十か条 日経谷島 宣之
	社内外の要員を組織し、保守・運用体制をきちんと整える。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	保守・運用のプロセスを整理し、ルールを決めておく。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	万一トラブルが起きてしまったときの連絡体制や対処方法を明確に決め、定期的に訓練を実施する。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	保守・運用のプロセスを改善する仕組み(評価指標の設定など)を用意する。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	保守・運用の生産性を高める技術基盤を用意する。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	経営トップや利用部門にシステムを定期的に評価をしてもらう。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	利用部門がシステムを使いこなせるように、研修サービスやマニュアルを充実する。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	「保守・運用部門は上流工程」と認識し、企画・開発部門に情報を積極的にフィードバックする。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	経営トップは保守・運用担当者の士気に気を配り、適時ローテーションを実施する。						システム保守・運用で失敗しない十か条 日経谷島 宣之
	システムはその会社のノウハウと価値そのものの						[serc-a:00435] (三輪さんメール)1
	ビジネス進化のために						[serc-a:00435] (三輪さんメール)2
	なぜ、保守エンジニア？						[serc-a:00435] (三輪さんメール)3
	新規システムへのリプレースなんだから保守エンジニアなんて要らん！？						[serc-a:00435] (三輪さんメール)4
	保守エンジニアに求められるもの(その1 価値と仕様)						[serc-a:00435] (三輪さんメール)5
	保守エンジニアに求められるもの(その2 素材とプログラミング構造 ※いい言葉が無いなあ)						[serc-a:00435] (三輪さんメール)6
	ドキュメンテーションは可視化の作りこみの一つ、これこそ保守の改善						[serc-a:00435] (三輪さんメール)7
	ソフト保守こそビジネス支援のエースの気概を持て						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)1
	常に保守プロセスの改善を心がけよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)2
	要件検討は案件の目的・ねらいを明確にせよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)3
	見積りはFP法の応用を考えよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)4
	影響調査はツール利用を検討せよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)5
	仕様作成は的確なドキュメントを残せ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)6
	ソフト作成は有効な手順を確立せよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)7
	テストはツール活用を前進させよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)8
	移行はOK確認の手順を明確にせよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)9
	障害対応は真の原因を追求せよ						ソフトウェア保守の心構え10か条(第1版) [serc-a:00400](上野さん案)10

タグBの定義

変更要求の発生・受付フェーズ

制度変更やサービス向上などの業務要因、システム能力向上などのシステム要因、及びシステム障害から発生した変更要求を受け付け、影響調査を関係部署に依頼する

影響調査フェース	受け付けられた変更要求に対して、システム変更の実現方法や、現行システム及び関連システムに与える影響について、範囲や規模、想定工数やリスク等を調査する。
変更実施判定フェース	変更実施判定フェース:影響調査の結果と優先順位、費用対効果、リスクを基にシステム変更の実施判定を行い、変更に関する契約を行う。
設計フェーズ	設計フェーズ:開発案件の設計を行い、設計内容を承認する。開発ベンダーとの開発進捗会議において、進捗状況、品質状況を石信忍し、必要に応じて改善指示を行う。
開発フェーズ	開発フェーズ:開発ベンダーが、開発環境を使用して開発とテストを実施する。開発ベンダーとの開発進捗会議で進捗状況/品質状況を確認し、必要なら改善指示すると共に、受け入れテスト/運用受け入れテスト/本番移行の準備を行う。
受け入れフェーズ	受け入れフェーズ:開発ベンダーが実施した開発作業の結果や変更後のシステスムを確認し、受け入れの可否を決定する。
受け入れテスト/運用受け入れテストフェース	受け入れテスト/運用受け入れテストフェース:受け入れテストと運用受け入れテストを実施し、本番に移/運用受け入れ行して良し1かを判断する。受け入れテストは受け入れテスト計画に従い開発ベンダーからの納品物が要求通りにできているかを確認する。運用受け入れテストは運用受け入れテスト計画に従い、運用上問題なくシステムが稼働するかを確認する。
本番移行作業フェーズ	本番移行作業フェーズ:本番移行計画に従い、関係者への周知を行い、本番環境ヘリスースする。正しく移行が行われたかどうかを確認し、成果物や関係ド、キュメントなどをライブラリに保管する。

第 22 年度(2012 年)SERC B グループ作業部会報告書

保守作業改善の基盤技術調査

B グループメンバー (順不動)

(株) NTT データ	峯村 圭介
個人研究員	福崎 哲郎
(財)経済調査会	押野 智樹
東芝ソリューション(株)	川上 康史
(株)日立ソリューションズ	木部 俊之
個人研究員	江尻 武志
(株)中電シーティーアイ	諸岡 隆司
(株)中電シーティーアイ	渡邊 将栄
個人研究員	石川 雅彦
(株) SRA	岸田 孝一
(株) SRA	方 学芬
(有)ウィルビーネットワーク	松尾 好博
個人研究員	小林 允

内容

I はじめに.....	70
II 研究報告.....	71
1 研究1:ソフトウェア進化・保守に関する最新動向の研究	71
1. 1 ICSM2012.....	71
1. 1. 1 はじめに	71
1. 1. 2 プログラム概要.....	71
1. 1. 3 プログラム詳細.....	72
1. 2 ICSE2013.....	82
1. 2. 1 はじめに	82
1. 2. 2 プログラム概要.....	82
1. 2. 3 プログラム詳細.....	84
1. 3 ICSM2013	87
1. 3. 1 はじめに	87
1. 3. 2 プログラム概要.....	87
1. 3. 3 プログラム詳細.....	88
2 研究2：保守業務の成果物と改善点・課題について	89
2. 1 はじめに.....	89
2. 2 保守業務の成果物	89
2. 3 保守業務の改善点・課題について.....	92
2. 4 最後に.....	92
III 活動報告	93
IV 今後の予定.....	93

I はじめに

ソフトウェア資産の増大とソフトウェアライフサイクルの長期化、またソフトウェア実行環境における外部モジュールへの依存度の増大により、ソフトウェア保守エンジニアの対応範囲は増大化複雑化の一途をたどっています。

それに加えて、一旦リリースされたソフトウェアは容易に廃棄されることは少なく、利用される方々の継続的な要望に応えるために絶えざる進化を要求されています。

このような状況に対処する方策として、わたしたちのグループは、保守作業改善の基盤技術に注目しました。保守作業改善の基盤技術は、ソフトウェア進化・保守を支援する、モデル、ツール、ベストプラクティスが含まれます。

保守作業改善の基盤技術は、支援環境と密接に結びついています。支援環境とは、豊富な計算機パワーや、インターネットと接続された環境を含みます。これらの基盤技術と支援環境を背景にしてわたしたちのグループでは、保守・開発に携わるソフトウェアエンジニアの能力を何倍にも強化(enhance)、増幅(enpower)し、自律したソフトウェアエンジニアの実践を通じてソフトウェアの進化・保守を促進するための研究を行うことを方針としています。

Ⅱ 研究報告

1 研究 1：ソフトウェア進化・保守に関する最新動向の研究

本章では ソフトウェア進化・保守に関する最新動向の研究成果を示す。最新動向の研究成果を得るために、ソフトウェア進化・保守に関する国際会議の動向を調査した。ソフトウェア進化・保守の最新動向を得るための国際会議としては、ICSM (International Conference on Software Maintenance), ICSE(International Conference on Software Engineering)などがある。まず最初に、ICSM2012を紹介し、発表された論文の中から本研究に関連の深いトピックスを示す。次いで、ICSE2013 を紹介し、発表された論文の中で本研究に関連の深いトピックスを示す。最後に、本年秋に開催予定の ICSM2013 のプログラムのの中から本研究に関連の深いトピックス及び論文を取り上げる。

1. 1 ICSM2012

1. 1. 1 はじめに

本章では、ICSM 2 0 1 2 を紹介し、発表された論文の中から本研究に関連の深いトピックスを示す..

1. 1. 2 プログラム概要

WWW ページ (<http://conferences.computer.org/icsm/>) によれば, ICSM2012 の概要は;

Since its start in 1983, ICSM (International Conference on Software Maintenance) has grown and developed into an international forum for software maintenance researchers and practitioners to examine key issues facing the software maintenance community. Participants from academia, government, and industry share ideas and experiences solving critical software maintenance problems.

1983 年に最初の会議が開催されて以来, ICSM は、産学に渡るソフトウェア保守研究者、実務者が参加し、ソフトウェア・メンテナンスコミュニティが直面している重要な問題を精査する場として成長発展してきた。産官学の参加者はソフトウェア・メンテナンス問題を解決するための経験やアイデアを共有する

2012 年のこの会議は、9 月 23 日から 9 月 30 日まで、イタリア・トレノで開催された。

本研究では、全体概要、プログラムを示した上で、注目を引くセッションについて紹介する。

	Morning		Afternoon	
Sunday 23	SCAM		SCAM	
Monday 24	SCAM	MESOCA	SCAM	MESOCA
Tuesday 25	ICSM plenary		ICSM parallel	ICSM parallel
	ICSM parallel	ICSM parallel		
Wednesday 26	ICSM plenary		ICSM parallel	ICSM parallel
	ICSM parallel	ICSM parallel		
Thursday 27	ICSM parallel	ICSM parallel	ICSM parallel	ICSM parallel
Friday 28	WSE	SSBSE	WSE	SSBSE
Saturday 29	SSBSE		SSBSE	
Sunday 30	SSBSE		SSBSE	

表 1 プログラム概要

(<http://selab.fbk.eu/icsm2012/index.php?p=program>)

プログラム概要は上の通りである。ICSM 本会議は 25 日から 27 日に行われ、会期前後には併設ワークショップが設けられている。

1. 1. 3 プログラム詳細

ICSM2012 のプログラム詳細は

http://selab.fbk.eu/icsm2012/index.php?p=detailed_program

に掲載されている。

初日、9 月 25 日は、オープニングの後、KeyNote が行われた。

KeyNote の Speaker はスイス Lugano 大学、イタリア Milano Bicocca 大学の

Mauro Pezze 教授で、タイトルは”From off-line to continuous on-line maintenance”

(<http://selab.fbk.eu/icsm2012/index.php?p=keynote>) によれば KeyNote の概要は；

Software is the cornerstone of the modern society, which can hardly tolerate failures and service discontinuity. At the same time, software systems are rapidly changing, and often rely on dynamically linked modules and services that may not be even available at design time. Classic off-line verification that require access to the whole software system, and stop-and-go maintenance that works off line badly adapt to

these new needs of modern software systems.

ソフトウェアは現代社会の基盤であり、故障やサービスの停止が容認できないものである。同時に、ソフトウェアシステムは変化が速く、しばしば動的にリンクされたモジュールと設計時点では利用さえできなかったサービスに依存している。全てのソフトウェアシステムへのアクセスを必要とする古典的なオフライン検証と オフラインで動く、少し進んでは止まるタイプの保守はこれら現代のソフトウェアシステムにまずい適合をする。

Self-healing technology addresses the new demands of software systems by moving some V&V activities from design to runtime. Self-healing systems can detect failures, diagnose, locate and correct faults fully automatically and at runtime, thus guaranteeing rapid recovery and resilience to software failures. In this talk, I discusses how systems can detect and heal failures and faults that are unknown at runtime without additional human intervention, identify intrinsic software redundancy as a great opportunity that can be exploited to automatically deal with emerging problems at runtime, and indicate how self-healing technology can impact on classic maintenance approaches.

自己治癒型の技法は、ある V&V アクティビティを設計時から実行時に移動することにより、ソフトウェアシステムの新たな要求に焦点をあてる。

自己治癒型システムは故障を検出し、診断を行い、全て自動で欠陥を特定し修正する。そしてそれ故に、実行時にソフトウェアの故障からの素早い回復を保証する。本キーノートでは、システムがどのように故障を検出し治癒するか、そしてシステムがどのように実行時の未知の欠陥を、人手を介することなく検出、治癒し、内在するソフトウェアの冗長さを、実行時に現れる問題を自動的に取り扱うために利用する大きな契機として特定するかを議論し、そしてどのように自己治癒型の技法が古典的なメンテナンスアプローチにインパクトを与えるかと示す。

セッション I は PROGRAM COMPREHENSION であり、3つの発表があった。

- Lijie Zou and Michael Godfrey(Univ. of Waterloo, CA).
An Industrial Case Study of Coman's Automated Task Detection Algorithm: What Worked, What Didn't, and Why
- Fleur Duseau, Bruno Dufour and Houari Sahraoui.
Vasco: A Visual Approach to Explore Object Churn in Framework-intensive Applications
- Seyed Mehdi Nasehi, Jonathan Sillito, Frank Maurer and Chris Burns.
What Makes a Good Code Example? A Study of Programming Q&A in StackOverflow

セッション II は、TESTING AND MAINTENANCE であり、3つの発表があった。

- Andrew Sutton and Marcin Zalewski. **Testing C++ Generic Libraries**
- Árpád Beszédes, Tamás Gergely, Lajos Schrettner, Judit Jász, László Langó and Tibor Gyimóthy. ***Code Coverage-Based Regression Test Selection and Prioritization in the WebKit System***
- Gabriele Bavota, Abdallah Qusef, Rocco Oliveto, Andrea De Lucia and Dave Binkley. ***An Empirical Analysis of the Distribution of Unit Test Smells and Their Impact on Software Maintenance***

セッション III は FAULT LOCALIZATION であり、2つの発表があった。

- Liang Gong, David Lo, Lingxiao Jiang and Hongyu Zhang. ***Interactive Fault Localization Leveraging Simple User Feedbacks***
- Chandan Rupakheti and Daqing Hou. ***Finding Errors from Reverse-Engineered Equality Models using a Constraint Solver***

セッション IV は MAINTENANCE ISSUES IN OO SYSTEMS であり、2つの発表があった。

- Benjamin Biegel, Fabian Beck, Willi Hornig and Stephan Diehl. ***The Order of Things: How Developers Sort Fields and Methods***
- Aditya Kumar, Andrew Sutton and Bjarne Stroustrup. ***Rejuvenating C++ Programs through Demacrofication***

セッション V は CHANGE IMPACT ANALYSIS であり、3つの発表があった。

- Neha Rungta, Suzette Person and Joshua Branchaud. ***A Change-Impact Analysis to Characterize Evolving Program Behaviors***
- Amir Reza Yazdanshenas and Leon Moonen. ***Fine-Grained Change Impact Analysis for Component-Based Product Families***
- Xiao Qu, Mithun Acharya and Brian Robinson. ***Configuration Selection Using Code Change Impact Analysis for Regression Testing***

セッション VI は ANALYSIS OF BUILD SYSTEMS であり、3つの発表があった。

- Andrew Neitsch, Kenny Wong and Michael W. Godfrey. ***Build System Issues in Multilanguage Software***

- Jafar Al-Kofahi, Hung Nguyen, Anh Nguyen, Tung Nguyen and Tien Nguyen. *Detecting Semantic Changes in Makefile Build Code*
- Roman Suvorov, Bram Adams, Meiyappan Nagappan, Ahmed Hassan and Ying Zou. *An Empirical Study of Build System Migrations in Practice: Case Studies on KDE and the Linux Kernel*

二日目、9月26日は Keynote で始まった。Keynoter は米 Delaware 大学 Computer and Information Sciences Department の Prof. Lori Pollock で、
タイトルは：

Leveraging Natural Language Analysis of Software: Achievements, Challenges, and Opportunities

ABSTRACT は (<http://selab.fbk.eu/icsm2012/index.php?p=keynote>) によれば；

Studies continue to report that more time is spent reading, locating, and comprehending code than actually writing code. The increasing size and complexity of software systems makes it significantly more challenging for humans to perform maintenance tasks on software without automated and semi-automated tools to support them, especially in the error-prone tasks. Thus, software engineers increasingly rely on software engineering tools to automate maintenance tasks as much as possible.

The program analyses that drive today's software engineering tools have historically focused on analyzing the program's data and control flow, dependencies, and other structural information about the program to uncover and prove program properties. Yet, a software system is more than just the source code and its structure. To build effective software tools, the underlying automated analyses need to use all the information available to make the tools as intelligent and useful as possible. By adapting natural language processing (NLP) to source code analysis, and integrating information retrieval (IR), NLP, and traditional program analyses, we can expect significant improvement in automated and semi-automated software engineering tools for many different software engineering tasks.

In this talk, I will overview research in text analysis of software and discuss our achievements to date, the challenges faced in text analysis, and the opportunities for text analysis of software in the future.

二日目の Keynote に続き、Session VII “TRACEABILITY” では、3つの論文発表があった。

- Patrick Mäder and Alexander Egyed. ***Assessing the Effect of Requirements Traceability for Software Maintenance***
- Hongyu Kuang, Patrick Mäder, Alexander Egyed, Achraf Ghabi, Hao Hu and Jian Lv. ***Do Data Dependencies in Source Code complement Control Dependencies for Understanding Requirements Traceability?***
- Nasir Ali, Zohreh Sharafi, Yann-Gael Gueheneuc and Giulio Antoniol. ***An Empirical Study on Requirements Traceability Using Eye-Tracking***

続いて,

Session VIII ”SOFTWARE CHANGES” では 3 つの論文発表があった.

- Tejinder Dhaliwal, Foutse Khomh, Ying Zou and Ahmed E. Hassan. ***Recovering Commit Dependencies for Selective Code Integration in Software Product Lines***
- Ameni Ben Fadhel, Marouane Kessentini, Philip Langer and Manuel Wimmer. ***Search-based Detection of High-level Model Changes***
- Yoshiki Higo and Shinji Kusumoto. ***How Often Do Unintended Inconsistencies Happen? Deriving Modification Patterns and Detecting Overlooked Code Fragments***

セッション IX “TEXTUAL ANALYSIS” では 2 つの論文発表があった.

- Anna Corazza, Sergio Di Martino and Valerio Maggio. ***LINSEN: An Approach to Split Identifiers and Expand Abbreviations with Linear Complexity***
- Abram Hindle, Christian Bird, Thomas Zimmermann and Nachiappan Nagappan. ***Relating Requirements to Implementation via Topic Analysis: Do Topics Extracted from Requirements Make Sense to Managers and Developers?***

セッション X “FAULT CORRECTION” では二つの論文発表があった

- Yuhua Qi, Xiaoguang Mao and Yan Lei. ***Making Automatic Repair for Large-scale Programs More Efficient Using Weak Recompile***
- Masao Ohira, Ahmed E. Hassan, Naoya Osawa and Kenichi Matsumoto. ***The impact of bug management patterns on bug fixing: a case study of Eclipse projects***

セッション X “CLONING” では, 3 つの論文発表があった.

- Hamid Abdul Basit, Usman Ali, Sidra Haque and Stan Jarzabek. *Things Structural Clones Tell that Simple Clones Don't*
- Gang Zhang, Xin Peng, Zhenchang Xing and Wenyun Zhao. *Cloning Practices: Why Developers Clone and What can be Changed*
- Manar Alalfi, James Cordy, Thomas Dean, Matthew Stephan and Andrew Stevenson. *Models are Code too: Near-miss Clone Detection for Simulink Models*

セッション XII “MAINTANABILITY” では、3つの論文発表があった。

- Aiko Yamashita and Leon Moonen. *Do Code Smells Reflect Important Maintainability Factors?*
- Tibor Bakota, Péter Hegedus, Gergely Ladányi, Péter Körtvélyesi, Rudolf Ferenc and Tibor Gyimóthy. *A Cost Model Based on Software Maintainability*
- Amjed Tahir and Stephen MacDonell. *A Systematic Mapping Study on Dynamic Software Metrics*

三日目 9月27日は、セッション XIII “REFACTORING” で始まった。

- Carlos Noguera, Andy Kellens, Coen De Roover and Viviane Jonckers. *Refactoring in the Presence of Annotations*
- Ali Ouni, Marouane Kessentini, Houari Sahraoui and Mohamed Salah Hamdi. *Search-based Refactoring : Towards Semantics Preservation*
- Napol Rachatasumrit and Miryung Kim. *An Empirical Investigation into the Impact of Refactoring on Regression Testing*

次に、セッション XIV “LIBRARY AND API EVOLUTION” では、3つの論文発表があった。

- John Businge, Alexander Serebrenik and Mark Van Den Brand. *Survival of Eclipse Third-party Plug-ins*
- Steven Raemaekers, Arie Van Deursen and Joost Visser. *Measuring Software Library Stability Through Historical Version Analysis*
- Hui Song, Gang Huang, Yingfei Xiong and Yanchun Sun. *Inferring the Data Access from the Clients of Generic APIs*

セッション XIV “LIBRARY AND API EVOLUTION”では3つの論文発表があった。

- John Businge, Alexander Serebrenik and Mark Van Den Brand. *Survival of Eclipse Third-party Plug-ins*
- Steven Raemaekers, Arie Van Deursen and Joost Visser. *Measuring Software Library Stability Through Historical Version Analysis*
- Hui Song, Gang Huang, Yingfei Xiong and Yanchun Sun. *Inferring the Data Access from the Clients of Generic APIs*

セッション XV “SPREADSHEET MAINTENANCE” では、二つの論文発表があった。

- Felienne Hermans, Martin Pinzger and Arie Van Deursen. *Code Smells in Spreadsheet Formulas*
- Sandro Badame and Danny Dig. *Refactoring meets Spreadsheet Formulas*

セッション XVI “BUG REPORTING” では、2つの論文発表があった。

- Ferdian Thung, David Lo, Lingxiao Jiang, Premkumar Devanbu, Lucia Lucia and Foyzur Rahman. *When Would This Bug Get Reported?*
- Rafael Lotufo, Zeeshan Malik and Krzysztof Czarnecki. *Modelling the ‘Hurried’ Bug Report Reading Process to Summarize Bug Reports*

セッション XVII “BUG AND WARNING MANAGEMENT” では二つの論文発表があった。

- Andre Hora, Nicolas Anquetil, Stéphane Ducasse and Simon Allier. *Domain Specific Warnings: Are They Any Better?*
- Mario Linares-Vasquez, Hoang Dang, Md Kamal Hossen, Huzefa Kagdi, Malcom Gethers and Denys Poshyvanyk. *Triaging Incoming Change Requests: Bug or Commit History, or Code Authorship?*

最後のセッション , セッション XVIII “CLUSTERING AND MODULARIZATION” では2つの論文発表があった。

- Kenichi Kobayashi, Manabu Kamimura, Koki Kato, Keisuke Yano and Akihiko Matsuo. *Feature-Gathering Dependency-Based Software Clustering Using Dedication and Modularity*

- Mathew Hall, Neil Walkinshaw and Phil McMinn. ***Supervised Software Modularisation***

この他、各セッションと並列で、ERA track と呼ばれるトラックの発表があった。

ERA Track とは

http://selab.fbk.eu/icsm2012/index.php?p=era_track

によれば;

The goal of the Early Research Achievements (ERA) track is to provide research and practitioners with a forum for presenting great ideas in early stages of research. The proposed ideas and promising work submitted to this track do not require to have been empirically evaluated. The topics of interest for this track are the same as the main research track and concern all the topics in the research and practice of software maintenance and evolution. Papers submitted to the ERA must not have been previously accepted for publication or submitted for review to another conference, journal, or book.

ERATrack の目標は、研究者、実務者に研究の初期段階における優れたアイデアを示すためのフォーラムを提供することである。このトラックに提出された、提示されたアイデア、将来的達成すべき仕事はエンピリカルに評価されている必要はない。このトラックで関心あるトピックは、リサーチトラックと同じであり、ソフトウェア保守および進化における研究及び実践の全てのトピックに関与する。ERA に提出された論文は既出のものであってはならない。

初日、最初の ERA track の”Software”では 4 つの論文発表があった。

- Enyi Tang, Linzhang Wang, Jianhua Zhao and Xuandong Li. ***Time-Leverage Points Detection for Time Sensitive Software Maintenance***
- Ju Qian and Xiaoyu Zhou. ***Inferring Weak References for Fixing Java Memory Leaks***
- Shuhei Kimura, Yoshiki Higo, Hiroshi Igaki and Shinji Kusumoto. ***Move Code Refactoring with Dynamic Analysis***
- James Hamilton and Sebastian Danicic. ***Dependence Communities in Source Code***

次の ERA track “Information” には 6 の論文発表があった。

- akayuki Omori, Hiroaki Kuwabara and Katsuhisa Maruyama. *A Study on Repetitiveness of Code Completion Operations*
- Dave Binkley, Dawn Lawrie and Christopher Uehlinger. *Vocabulary Normalization Improves IR-Based Concept Location*
- Erik Kouters, Bogdan Vasilescu, Alexander Serebrenik and Mark Van Den Brand. *Who's who in Gnome: using LSA to merge software repository identities*
- Philips K. Prasetyo, David Lo, Palakorn Achananuparp, Yuan Tian and Ee-Peng Lim. *Automatic Classification of Software Related Microblogs*
- Ferdian Thung, David Lo and Lingxiao Jiang. *Detecting Similar Applications Leveraging Collaborative Tagging*
- Shaowei Wang, David Lo and Lingxiao Jiang. *Inferring Semantically Related Software Terms and Their Taxonomy By Leveraging Collaborative Tagging*

三日目の ERA Track “History”では、6 の発表があった。

- Girish Maskeri Rama, Deepthi Karnam, Srinivas Padmanabhuni and Sree Aurovindh Viswanathan. *Version History Based Source Code Plagiarism Detection in Proprietary Systems*
- Aseel Hmood, Mostafa Erfani, Iman Keivanloo and Juergen Rilling. *Applying technical stock market indicators to analyze and predict the evolvability of open source projects*
- Shinpei Hayashi, Takayuki Omori, Teruyoshi Zenmyo, Katsuhisa Maruyama and Motoshi Saeki. *Refactoring Edit History of Source Code*
- Shusi Yu. *Retrieving Software Maintenance History with Topic Models*
- Jerod Wilkerson. *A Software Change Impact Analysis Taxonomy*
- Foutse Khomh, Hao Yuan and Ying Zou. *Adapting Linux for Mobile Platforms: An Empirical Study of Android*

その他のトラック、セッション、ワークショップ

併設トラックとして Industry Track が開催されている。また、ワークショップ、ツールデモンストレーション、PhD セッションなどが開催されている。

ERA track と併設で開催された Industry track とは、

(http://selab.fbk.eu/icsm2012/index.php?p=industry_track)によれば；

The Industry Track of the ICSM 2012 conference aims to bring together people from both academia and industry in a venue that highlights practical and real world studies of software maintenance. This track aims to foster mutually beneficial links between those engaged in scientific research and practitioners working to make software maintenance efficient. We are interested in results (both good and bad), obstacles, and lessons learned associated with applying software maintenance practices. Experiences from practitioners provide crucial input into future research directions and allow others to learn from successes and failures.

For the industry track, we invite submissions of state-of-the-art descriptions, state-of-the-art practice and experience reports, and survey reports from real-world projects and industrial experiences. If you apply in an industrial context a method, model or tool, which you know was earlier presented at ICSM or other software engineering conference, we also warmly encourage you to submit a paper to this track.

1. 2 ICSE2013

1. 2. 1 はじめに

本章では、ICSE2013 を紹介し、発表された論文の中から本研究に関連の深いトピックを示す。

内容については WWW ページ <http://2013.icse-conferences.org/> に依った。

1. 2. 2 プログラム概要

ICSE2013 の概要については、WWW ページのトップページに詳しい記述がある。

The International Conference on Software Engineering, [ICSE](#), provides programs where researchers, practitioners, and educators present, discuss, and debate the most recent innovations, trends, experiences, and challenges in the field of software engineering.

ICSE は研究者、実務家教育者に対して、ソフトウェアエンジニアリングの分野における最も最近の技術革新、主潮流、経験、課題を討議、議論、プレゼンテーションするプログラムを提供する

ICSE 2013, the 35th in the conference series, encourages contributors from academia, industry, and government to share leading-edge software engineering ideas with inspirational leaders in the field. All events are at the [Hyatt Regency San Francisco](#), right in the heart of the Embarcadero District, in view of the San Francisco Bay and the Golden Gate bridge. Please join us for what promises to be a memorable ICSE!

35回目を迎える ICSE2013 は、サンフランシスコで開催された。

Opportunities for professional engagement include workshops, tutorials, demonstrations, posters, exhibits, paper tracks on research, education and software engineering in practice, as well as a set of co-located events. Students, as the lifeblood of the field, will be prominent at ICSE 2013 not only as contributors to the conventional opportunities, but also through the ACM Student Research Competition (SRC) and the worldwide Student Contest on Software Engineering (SCORE). The main conference (May 22–24) will have multiple tracks that intermix research, education, practice, demonstrations, posters, new and emerging

ideas, and keynotes. Tutorials, workshops, and co-located events will be held before (May 18–21) and after (May 24–26) the main conference.

メインコンファレンスは5月22日から24日にかけて複数のトラックで構成される。複数のトラックは研究、教育、実践、デモ、ポスター、新しいアイデア、キーノートが混在している。チュートリアル、ワークショップ併設イベントは会期前の5月18日から21日、会期後の5月24日から26日にかけて開催された。

MAIN CONFERENCE (Conference starts at 8:15 a.m.)			
	Wed. May 22	Thu. May 23	Fri. May 24
Keynotes	Pam Samuelson	Tony DeRose	Linda Northrop
AM Break (10:00-10:30)			
2.1	Composition	Code Analysis	Reliability
2.2	Testing	Big Data	Analysis Studies
2.3	Formal Analysis	Search-Based SE	Programming Support
2.4	SEIP	SEIP	SEIP
2.5	NIER	NIER	SEIP
Lunch (12:00-1:30)			
3.1	Adaptation	Debugging	Security & Privacy
3.2	Test Case Generation	Process	Empirical Studies
3.3	Formal Specifications	Performance	Program Repair
3.4	SEIP	SEIP	Tools
3.5	NIER	NIER	
3.6	SEE	SEE	DEMO
PM Break (4:00-4:30)			
4.1	Applications	Bug Prediction	Closing Plenary
4.2	Test Case Selection	Product Lines	
4.3	Analysis	Req. Engineering	
4.4	SEIP	SEIP	
4.5	SEIP	DEMO	
4.6	SEE	SEE	

ICSE2013 本会議プログラム概要

(<http://2013.icse-conferences.org/content/icse-2013-glance.html>)

1. 2. 3 プログラム詳細

ICSE のセッションは広範囲に及ぶため、ここではすべての論文を挙げることはできない。セッションの一覧のみを示す。

- Composition
- Adaptation
- Apps
- Testing
- Test-Case Generation
- Test-Case Selection
- Formal Analysis
- Formal Specification
- Analysis
- Code Analysis
- Debugging
- Bug Prediction
- Big Data
- Process
- Product Lines
- Search-Based SE
- Performance
- Requirement Engineering
- Reliability
- Security and Privacy
- Analysis Studies
- Empirical Studies
- Programming Support
- Program Repair
- Tools
- Keynotes
- Technical Debt: Past, Present, and Future
- Agile and Distributed Practices
- Software Architecture
- Metrics and Evaluation
- Mini-Tutorial
- Case Studies
- Testing

- Bug Detection
- Problem-Based and Studio Learning
- Teaching Introductory Software Engineering
- Panel/Town Hall Discussion of SE 2004 Revisions
- Advanced Software Engineering Education
- Dependability Perspectives
- Supporting Tomorrow's Developer
- Collaborative Development
- Alternative Modeling
- Posters
- Formal Demonstrations1
- Formal Demonstrations2
- Short Papers
- Posters
- Program Analysis
- Debugging
- Process and Maintenance
- Models and Requirements
- Developers and Users
- Tutorial Summaries
- Workshop Summaries
-

Reliability セッションの中に, Safe software updates via multi-version execution というタイトルの論文がある. この論文のタイトルと ABSTRACT をここに掲載する.

タイトル : Safe Software Updates via Multi-version Execution

(複数のバージョン実行による安全なソフトウェア更新)

Abstract—Software systems are constantly evolving, with new versions and patches being released on a continuous basis. Unfortunately, software updates present a high risk, with many releases introducing new bugs and security vulnerabilities. We tackle this problem using a simple but effective multiversion based approach. Whenever a new update becomes available, instead of upgrading the software to the new version, we run the new version in parallel with the old; by carefully coordinating their executions and selecting the behavior of the more reliable version when they diverge, we create a more secure and dependable multi-version

application.

We have implemented this technique in a prototype system targeting multicore processors, and show that it can be applied successfully to several security-critical applications, such as `lighttpd` and `redis`.

Keywords-`software updates`, `multi-version execution`, `security vulnerabilities`

概要: 一連の基盤に対して新たなバージョンとパッチがリリースされるに伴い、ソフトウェアシステムは絶えず進化している。不幸にも 多くのリリースが新たなバグとセキュリティ脆弱性を引き起こすため、ソフトウェアの更新には高いリスクが伴う。我々は、シンプルだが効果的なマルチバージョンベースのアプローチを使ってこの問題に取り組む。新たな更新が利用可能になる時にはいつでも、そのソフトウェアを新しいバージョンにアップグレードする代わりに、新しいバージョンと古いバージョンを並行して走らせる。それらの実行を注意深く調整し、二つのバージョンが異なる結果を示す時により信頼性のあるバージョンの振る舞いを選択することにより、我々はよりセキュアでより信頼性の高いマルチバージョンアプリケーションを生成する。

我々は、この技法をマルチコアプロセッサをターゲットとするプロトタイプシステムに実装した。そして、そのプロトタイプがうまく適用され、`lighttpd` と `redis` のような、セキュリティ上クリティカルなアプリケーションに適用した。

キーワード: ソフトウェア更新, マルチバージョン実行, セキュリティ脆弱性

1. 3 ICSM2013

1. 3. 1 はじめに

本章では、ICSM2013 のプログラム概要と詳細について紹介する。

1. 3. 2 プログラム概要

ICSM2013 は 2013 年 9 月 22 日から 28 日まで、オランダ Eindhoven で開催される。¹ Web のトップページ <http://icsm2013.tue.nl/index.html> によれば、開催趣旨は；

ICSM is the premiere international venue in software maintenance and evolution, where participants from academia, government, and industry meet and share ideas and experiences for solving critical software maintenance problems.

また、ICSM 2013 では

- ・ Research Track
- ・ Early Research Achievements(ERA) Track
- ・ Industry Track
- ・ Tools Track
- ・ Doctoral Symposium

の 5 つの Track が開催されることになっている。加えて ICSM2013 では、5 つの併設イベントが開催される。

- ・ 13th IEEE International Working Conference on Source Code Analysis and Manipulation(SCAM)
- ・ 15th IEEE International Symposium on Web Systems Evolution (WSE)
- ・ 7th International Symposium on the Maintenance and Evolution of Service-oriented and Cloud-Based Systems(MESOCA)
- ・ 1st IEEE Working Conference on Software Visualization(VISSOFT)
- ・ 1st International Workshop on Communicating Business Process and Software Models ; Quality, Understandability and Maintainability(CPSM)

¹ 本稿執筆時点で、未開催である.ここでは Web Page の内容に基づき記述する.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Sept 22	Sept 23	Sept 24	Sept 25	Sept 26	Sept 27	Sept 28
SCAM		ICSM			VISSOFT	
	MESOCA				WSE	
	CPSM					
SCAM Banquet		ICSM Reception	ICSM Banquet		VISSOFT Banquet	

ICSM2013 プログラム概要

(<http://icsm2013.tue.nl/Schedule/index.html>)

1. 3. 3 プログラム詳細

ICSM の Research Track は、初日に 5、二日目に 4、最終日に 5 Track の計 14 Tracks が開催される予定になっている。

第 1 日

- Software Comprehension
- Software Authorship
- Reverse Engineering
- Smells and Anti-patterns
- Refactoring

第 2 日

- APIs
- Dependencies
- Context
- Code Cloning

最終日

- Runtime Analysis
- Traceability
- Fault and Defect Management
- Feature Location
- Testing

2 研究 2：保守業務の成果物と改善点・課題について

中電シーティーアイ 諸岡隆司、渡邊将栄

2.1 はじめに

ソフトウェア保守は、ソフトウェアライフサイクルにおける主プロセスの一つで、出荷後のソフトウェアを常に正常に稼働させることを目的として、環境の変化に対応するための変更を行ったり、ソフトウェアの欠陥を修正したり、性能改善等を行う重要なプロセスである。また、ソフトウェアにかかる総コストの 50%～80%を消費するといわれており、当社においてもソフトウェア保守は主要な事業の一つである。

そのような重要な位置付けにある保守業務について、当社のあるプロジェクト（改良開発）の評価をしたので、その結果を報告する。

2.2 保守業務の成果物

保守業務の定義やスコープは、お客さまの運用や保守環境等の違いにより様々である。今回の評価対象の保守業務のスコープを確認するために、SLCP-JCF のプロセスに保守業務での作成成果物をマッピングした。

SLCP-JCF は、共通フレームとも呼ばれ、発注者と受注者の間で、開発作業に対する相互誤解がないように、様々な作業内容の詳細を規定したものである。

マッピングした結果より、成果物の傾向と見解について下表に示す。

表 1 SLCP-JCF のプロセスに成果物をマッピングした結果

成果物の傾向	見解
「企画」「要件定義」「システム要件定義」の成果物が少ない	<ul style="list-style-type: none"> ・既にシステムとしては開発されて運用されているため、「企画・要件定義」を実施し直す必要はないとの判断である。（ただし、変更量が多い場合は、開発業務と同様に「企画・要件定義」を実施する） ・「システム要件定義」については、お客さま側で実施し、仕様書として提示されるケースが多い。
「システム適確性確認テスト」の成果物が少ない	<ul style="list-style-type: none"> ・V字開発を想定すると「システム要求定義」の成果物が少ないことに関連して「システム適確性確認テスト」の成果物も少なくなっている。 ・ただし「システム適確性確認テスト」の内容については、「結合テスト」で実施しており、「総合テスト」としての成果物は作成していないのが実状である。
「ソフトウェア受入」「運用テスト」の成果物が少ない	<ul style="list-style-type: none"> ・「ソフトウェア受入」「運用テスト」は、お客さま側の担当で、スコープに含まれていないケースが多い。
「ソフトウェア要求定義」の成果物が多い	<ul style="list-style-type: none"> ・改良開発を行うシステムは、現行システムとして稼働しており、改良することでシステムが停止するなどの障害は、発生してはならないため、「ソフトウェア要求定義」の設計を重視して行うことで、品質確保の1つの策としている。
「単体テスト」に対応する設計書が詳細設計工程で作成していない	<ul style="list-style-type: none"> ・「単体テスト」のテストケースは、基本設計の設計書とプログラムソースから作成されており、対応する設計書（プログラム設計書等）がないことの課題感はあるものの、作成する作業負荷等を考慮すると作成できていないのが現状である。

「企画」「要件定義」の超上流工程、「システム適確性確認テスト」「ソフトウェア受入」「運用テスト」の総合テストの成果物が少ないことは、保守業務のために工程を省いているのではなく、保守業務の内容によって、各プロセスの実施判断をおこなっているため、場合によっては一部機能の要件定義や負荷テスト等の総合テストを実施する必要がある。保守業務の内容に応じたプロセス設計を行うスキルが必要である。

プログラム設計書を作成することについては、開発業務でもプログラム設計書を作成しないプロジェクトもあるため、効率を考慮した上で必要性を確認していきたい。

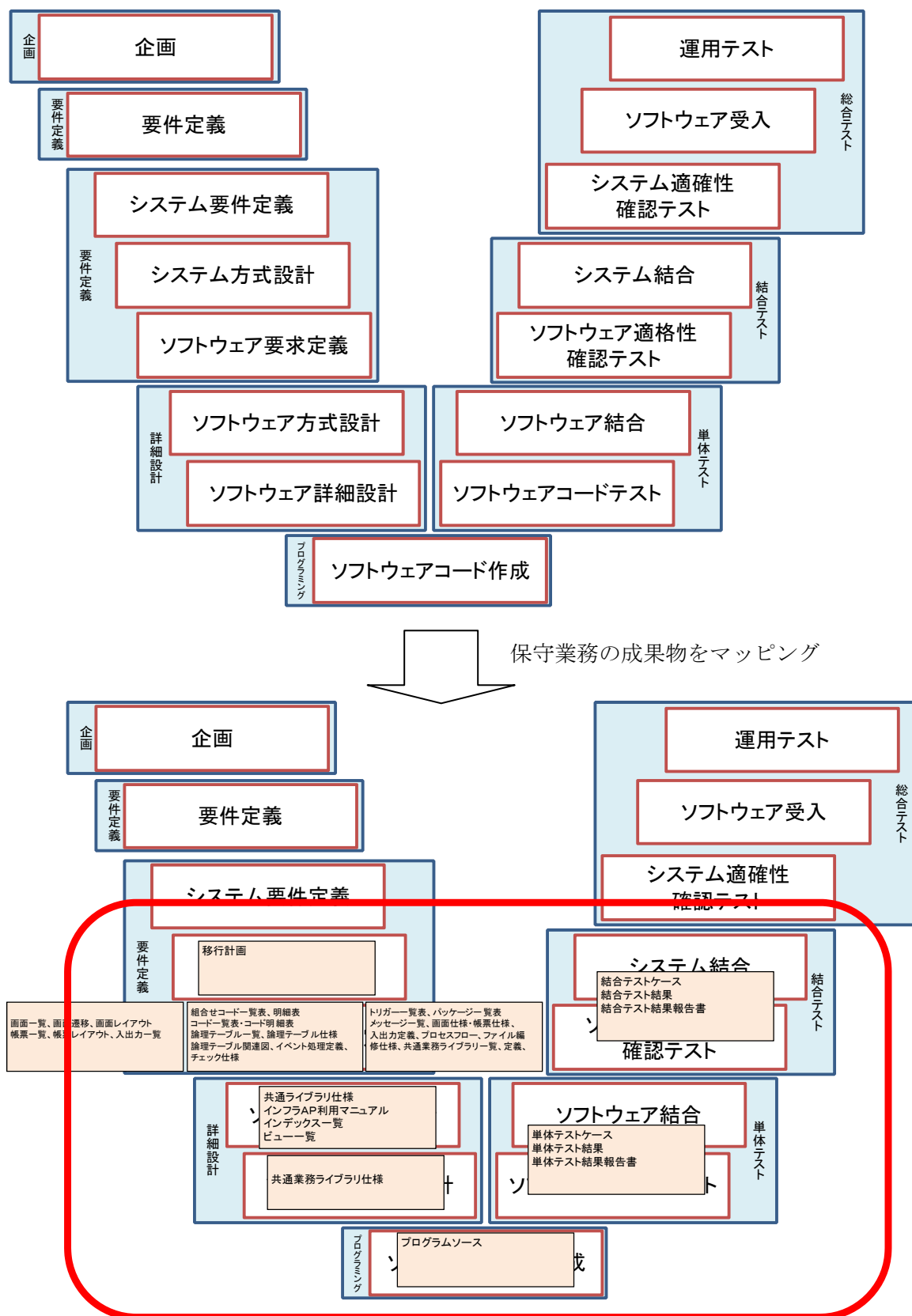


図 1 SLCP 工程定義と保守業務の成果物

2.3 保守業務の改善点・課題について

保守業務については、定常的に生産性、品質等に関する改善検討を行い実施している。その内容や現状の課題感について示す。

(1) 知識・ノウハウの共有

保守業務では、技術の進歩があまり見られず、徒弟制度的な要員育成に頼っている傾向にあるため、ノウハウの共有が課題とされている。そのため、部署内で保守で必要なスキルを洗い出し、担当者のスキル評価、教育を実施する仕組みを3年かけて構築した。

対象システムの業務知識の習得についても取り組んでおり、マニュアルの作成、有識者による勉強会の開催等を実施している。また、障害対応マニュアルや問合せ対応マニュアルなども整備して、障害の再発防止に役立てている。

(2) ツールの利用促進

開発ツールについては、基本的に開発時に使用していたツールを引き続き利用している。保守業務でも必要性を検討し、mudflap (バッファオーバーフロー等の検出) 等のツールを採用しているが、まだまだ利用が少ないと感じているため、より高い生産性向上、品質向上に向けてツールの利用促進を行っていききたい。

(3) レビュー方法の工夫

設計書やプログラムソースの品質向上策として、有識者によるレビューを実施しており、ある程度の成果を上げている。レビュー方法についてはレビュー体制の計画やチェックリストの作成等を行っているが、作業時間としてはかなりの時間を要しているため、レビュー方法の改善を図っていききたい。

(4) メンテナンスドキュメントの整理

新規開発プロセスのドキュメントは多岐にわたり、少人数で実施する保守プロセスで、全てのドキュメントをメンテナンスしていくのは非効率である。そのため、開発時に作成した設計書をもとに、保守業務で必要なドキュメントを整理し、設計書の名称等の統一を図る等の整備を進めている。(H25.9 末完了予定)

2.4 最後に

今回の活動で、保守業務の成果物や、ここ数年実施してきた改善活動等を整理したため、この内容をもとに今後の改善点等を洗い出し、来年度の活動つなげていきたい。

Ⅲ 活動報告

本グループの活動は

- (1) キックオフミーティングでの討議
- (2) 月例会による討議
- (3) ML 上によるオンラインでの討議
- (4) 夏期に開催する合宿または暑気払い
- (5) 報告書作成後の打ち上げ

などによって構成される。

今年度の活動は

- ・キックオフミーティングにおいて、参加メンバーによる討議が行われた。
- ・ML を使って、討議や活動の中間報告が行われた。
- ・月例会、夏期のイベントは未開催に終わった。
- ・打ち上げについては、開催は未定。

という状況となった。

Ⅳ 今後の予定

昨年度の報告では、今後の予定として以下を挙げた。

- ・ソフトウェア・ツールの研究
 - ◇ ソフトウェア保守・進化に有用なツールを発掘し、評価する
- ・ソフトウェア・サステナビリティの研究
 - ◇ ソフトウェア・サステナビリティの動向調査
 - ◇ ソフトウェア・サステナビリティの概念を把握する活動
- ・フォーラムの開催
 - ◇ 参加者と研究員とのディスカッション要素をより多く取り入れたフォーラムの開催
- ・その他、ソフトウェア進化・保守を支える考え方の調査
 - ◇ ソフトウェア保守開発以外の分野に関して、ソフトウェア保守・進化と共通の性格を持つ、考え方や技法の調査、把握

これらの方針を引き続き掲げていくか、または方向修正や新規追加を行うかは、グループ内での今後の議論によって決めたい。

第 22 年度 S E R C 報告書

グループ名：

「10年後のソフトウェア保守を考える」

S E R C－C グループ

研究テーマ：

「ソフトウェア保守プロセス標準化への取り組み」

参加メンバー：

(株)アイ・ティ・フロンティア
中央コンピューター(株)
アイエックス・ナレッジ(株)
アイエックス・ナレッジ(株)
東芝ソリューション(株)

丸山 陽一
伊藤 順一
岡田 浩
井瀬 英晶
佐井 由美子

2013年9月20日

目次

1. はじめに	96
2. ソフトウェア保守プロセス標準化への取り組み	97
3. ソフトウェア保守プロセス実装チェックリスト作成と解説	99
3.1 チェックリストの対象範囲	99
3.2 チェックリストのキーワード	100
3.3 チェックリストの作成	102
3.4 チェックリストの項目	102
3.5 チェックリスト内容の補足説明	104
4. ソフトウェア保守プロセス実装チェックリストの各社評価報告	114
4.1 各社評価報告の概要	114
4.2 各社評価報告の構成	115
4.3 A社ソフトウェア保守のチェック結果および評価報告	116
4.4 B社ソフトウェア保守のチェック結果および評価報告	120
4.5 C社ソフトウェア保守のチェック結果および評価報告	122
4.6 D社ソフトウェア保守のチェック結果および評価報告	124
5. おわりに	126
5.1 今年度の総括	126
添付資料 表 3-3 ソフトウェア保守プロセス実装チェックリスト (1/3)	128
添付資料 表 4-3 A社ソフトウェア保守プロセス実装チェックリスト (1/3) ..	131
添付資料 表 4-4 B社ソフトウェア保守プロセス実装チェックリスト (1/3) ..	134
添付資料 表 4-5 C社ソフトウェア保守プロセス実装チェックリスト (1/3) ..	137
添付資料 表 4-6 D社ソフトウェア保守プロセス実装チェックリスト (1/3) ..	140

【参考文献】

- ・ソフトウェア技術—ソフトウェアライフサイクルプロセス—保守 JIS X 0161:2008
- ・角田雅照 門田暁人 松本健一 押野智樹「受託開発ソフトウェアの保守における作業効率の要因」コンピュータソフトウェア, Vol. 29, No. 3 (2012), pp. 157-163
- ・増井和也／弘中茂樹／馬場辰男／松永真 「～IS014764 による～ソフトウェア保守開発」ソフト・リサーチ・センター 2007 年 10 月
- ・ソフトウェア製品の品質—第 1 部：品質モデル JIS X 0129-1:2003
- ・保田勝道／奈良隆正 「ソフトウェア品質保証入門」 日科技連出版社 2008 年 4 月
- ・独立行政法人 情報処理推進機構 (IPA)「共通フレーム 2013」 2013 年 3 月
- ・一般財団法人経済調査会 「平成 24 年度ソフトウェア保守に関する調査票」 2012 年 11 月

1. はじめに

Cグループは、2003年より「保守のアウトソーシングを考える」をメインテーマとして保守現場に密着したサブテーマを毎年選択し研究活動を継続してきた。しかし、活動も9年経過する中で経済環境も大きく変化したことから、次の10年へ向けてメインテーマ「10年後の保守を考える」を新たに設定した。

2012年は毎回フォーラム形式を主体とした活動にしたいと考え、ゲストスピーチの講演を中心にフォーラムを2回開催し、様々な観点からメインテーマについて議論を行なった。その結果、若手保守技術者への知識や技術の伝承が課題として残った。

そこで今年度は、「若手保守技術者に知識や技術を早く確実に伝承するにはどうすればよいか？」をテーマに、ソフトウェア保守プロセスの標準化（明文化）を進めれば、効率よく伝承できるのではないかと考え、チェックリストによる標準化推進の研究に取り組んだ。

また、ベテラン保守技術者が定年退職により抜けたため、「ソフトウェア技術－ソフトウェアライフサイクルプロセス－保守 JIS X 0161:2008」を理解していないメンバーの比率が上がったため、この規格の勉強を兼ねてソフトウェア保守プロセス実装チェックリスト作成に取り組んだ。

この報告書では以降、文章の読みやすさを考慮し、次の項目を略称で表記する。

- ・ソフトウェア技術－ソフトウェアライフサイクルプロセス－保守 JIS X 0161:2008 は、「JIS 規格」と略す。
- ・ソフトウェア保守プロセス実装チェックリストは、「チェックリスト」と略す。

2. ソフトウェア保守プロセス標準化への取り組み

複雑化、多様化して行くソフトウェア保守環境において、どの様にして若手エンジニアに早く、確実にノウハウを伝承できるのか？ この課題は、有識者が定年退職を迎え保守技術者の低年齢化が進む現状を踏まえ、Cグループが前年度より取り組んだ研究課題であった。

2年目の研究テーマを検討するため、2012年11月に行われたソフトウェア・メンテナンス研究会の合宿に臨み、そこで東洋大学 総合情報学部 助教授である角田氏の基調講演「ソフトウェア保守に影響する要因の分析」を拝聴し、今後の研究テーマとなる一つのヒントを得ることができた。

講演において角田氏から、「保守プロセスの標準化状況と技術者あたりの保守量に関連性があり、保守プロセスを標準化することにより、技術者あたりの保守量がおおむね8倍(少なく見積もると2倍、多く見積もると35倍)程度改善することが期待される」とし、保守プロセスの標準化が、生産性の向上に大きく寄与することが研究結果にて明らかとなったことをご紹介いただいた。

Cグループは、角田氏の基調講演を受け、各企業のソフトウェア保守の標準化状況を測定し、若手エンジニアにノウハウを伝承するための各社取り組みを可視化することを目的に、ソフトウェア保守プロセスとしてのJIS規格をチェックリスト化し、Cグループ参加メンバーによるチェック実施結果の評価の結果にて得ることができるであろう推奨すべき事例を取り上げたいと考えた。

推奨すべき事例は、ソフトウェア・メンテナンス研究会を通じ、多くの保守現場において取り入れ、現在の取り組みを補完していただけるような情報としてCグループより提供していきたい。

JIS規格にはソフトウェア保守プロセスとして6つのアクティビティが定義されており、Cグループでは今後この6つのアクティビティ毎に保守プロセスをチェックリスト化し評価結果を報告して行きたいと考えている。

【ソフトウェア保守プロセスの6つのアクティビティ】

- ① 保守プロセス実装
- ② 問題の把握および修正分析
- ③ 修正の実施

- ④ 保守レビューおよび受入
- ⑤ 移行
- ⑥ 廃棄

Cグループでは、本年度の研究として「①保守プロセスの実装」に焦点を当て研究を進めた。

3. ソフトウェア保守プロセス実装チェックリスト作成と解説

Cグループでは、チェックリストを作成するにあたり、JIS 規格を参考にした。この JIS 規格からソフトウェア保守プロセス実装のチェックに必要な項目を洗い出し、Cグループメンバー独自の解釈を加えチェックリストとしてまとめた。この章では、はじめにチェックリスト作成手順を説明し、次にチェックリストの見方と記入方法を説明し、最後にいくつかのチェックリスト項目について解説する。

この章では、文章の読みやすさを考慮し、Cグループメンバーを「メンバー」、ソフトウェア保守プロセスを「保守」と略して表記する。

3.1 チェックリストの対象範囲

JIS 規格は、保守で実行すべきタスクを網羅した形となっている。この JIS 規格は保守の多様性が考慮され、時間（工程）にも成果物（文書）にも依存していない。

今年度は、JIS 規格の「プロセス実装アクティビティ」についてチェックリストの作成に取り組んだため、JIS 規格の箇条「5.1 プロセスの実装」、「6 実施時の考慮事項」「7 ソフトウェア保守の戦略」を対象とした。保守の多様性については、表 3－1「ソフトウェア保守プロセス多様性の例」を参照のこと。

表 3－1 ソフトウェア保守プロセス多様性の例

カテゴリ	多様性の例
システム形態	企業の規模や業種・業務による多様性。 金融業、製造業、通信事業、リース業などの違い。営業販売（請求）・在庫（受発注）・人事経理・生産・物流などによる違い。
ソフトウェア形態	エンタープライズ系・パッケージ系・組み込み系による多様性。 オンライン中心、バッチ中心、外部接続などの違い。
保守契約	契約形態（作業請負・一括請負・保守案件毎の時間精算）、契約期間、契約人数（1人～大規模組織）による多様性。
保守の規模	保守対象のソフトウェア数や量による人数の多様性。1人で複数のソフトウェア保守を行なうなど、担当するソフトウェアの規模も違う。
保守のタイプ	是正、緊急、予防、適応、完全化のいずれか、または全て。JIS 規格では保守のタイプは 5 種類に分類されている。
その他	複数の保守案件が並行して実施される。保守案件発生の都度、影響度や緊急度により優先度が変化する。

メンバーは、既に保守を実装している状況で作業を行なっていることから、新規に保守を実装する際のチェックリストではなく、保守作業を続けている状態でどの程度標準化が進んでいるかのチェックリストを目指すことにした。図3-1「ソフトウェア保守プロセス実装のケースの例」を参照のこと。

Aについては、別組織で新規システムの開発が行なわれ、新規に保守が立ち上がるケースである。(ソフトウェアについては知識がない状態からのスタートすることを想定)

Bについては、別組織で保守を行なっていたが、別組織の理由により、新規に保守が立ち上がるケースである。(保守会社の倒産など、保守業務を引き取ることを想定)

Cについては、保守を継続中であるが、別組織で新規システムの開発が行なわれ、新たに保守対象のソフトウェアが追加になるケースである。(クライアントサーバー系の保守からWeb系の保守へ切り替わることを想定)

Dについては、保守を継続中であり、今後も保守が継続するケースである。(アーキテクチャの変更はなく、保守の中でソフトウェア開発プロセスを行なっていることを想定)

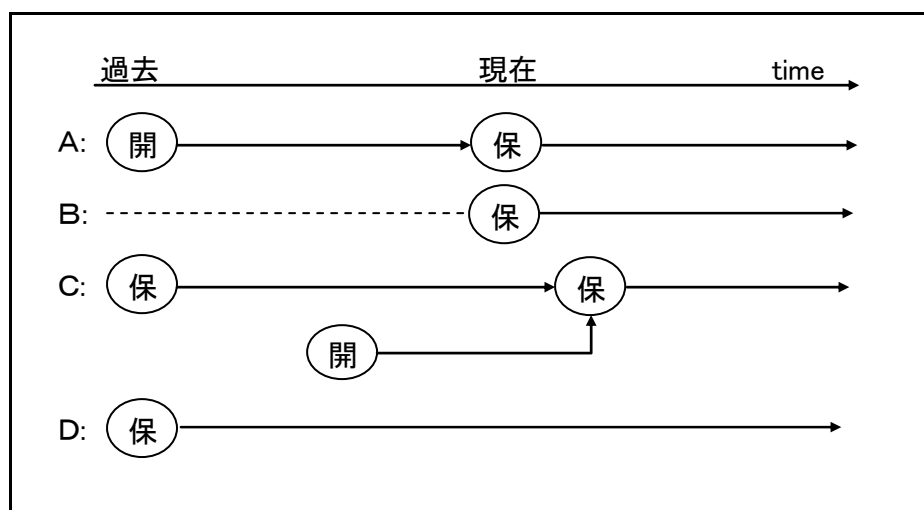


図3-1 ソフトウェア保守実装のケースの例

3.2 チェックリストのキーワード

JIS規格「5.1 プロセスの実装」の「5.1.5 出力」には11種類の出力が記述されている。11種類の出力を次に示す。

- ・保守計画
- ・教育訓練
- ・保守手続き

- ・プロジェクト管理計画
- ・問題解決手続き
- ・測定計画
- ・保守マニュアル
- ・利用者フィードバック計画
- ・引継ぎ計画
- ・保守性アセスメント
- ・構成管理計画

最初に、この出力を文書名とし、メンバーの保守現場での文書の有無、文書の内容からチェック項目の洗い出しを試みた。しかし、文書名が該当するものが少なく、チェック項目を集めることができなかった。

理由としては、メンバーの保守現場により、文書名は同じでも内容が違っていたり、文書名が違って内容が似ていたりしたからである。例えば、保守計画書と言う文書がない。例えば、プロジェクト管理計画書の中に品質計画の項目がある。例えば、品質計画書の中に教育訓練の項目があったりした。少人数による作業請負契約でソフトウェア保守を行っている場合は、顧客側に保守実装時に必要な文書があり、自分たちは参照するだけかも知れない。

保守の多様性による分類分けを不要とし、汎用的なチェックリストにするために、文書名と捉えるのを止め、保守実装に必要なキーワードと捉えることにした。メンバーが考えたキーワードの連鎖イメージを図3-2「JIS規格の出力イメージ」に示す。

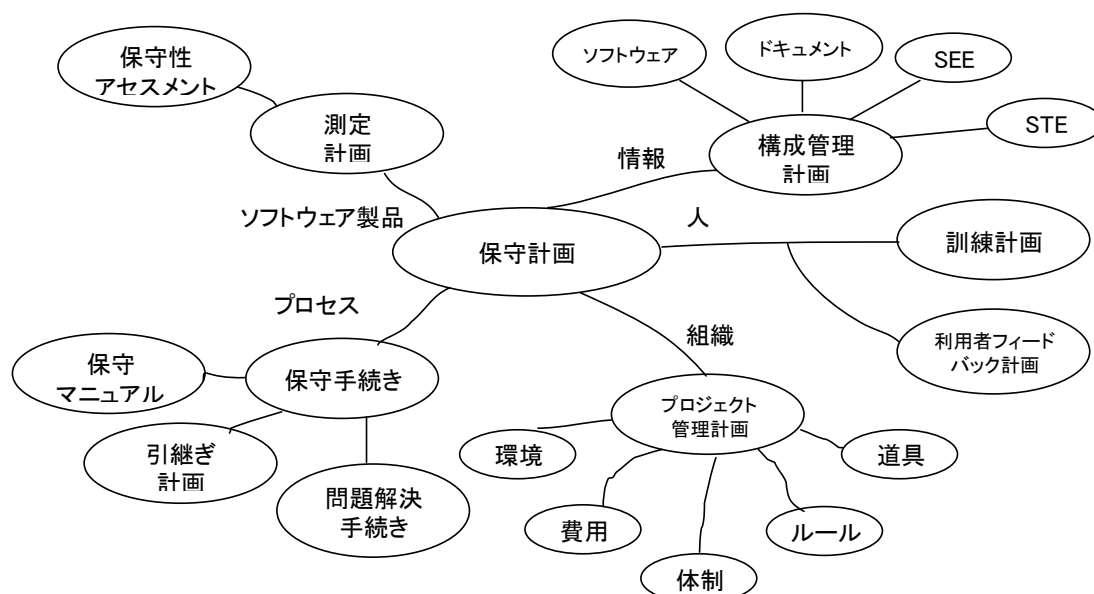


図3-2 「JIS規格の出力イメージ」

3.3 チェックリストの作成

チェックリストのキーワードの洗い出しにあたっては、JIS 規格の箇条「5.1 プロセスの実装」、「6 実施時の考慮事項」「7 ソフトウェア保守の戦略」をテキストに起こし、キーワードの文字列検索を行ない、検索されたキーワードを含む文を抽出し、表計算ソフトのシートに取り込んだ。キーワードによる検索した結果を都度シートに取りこんだため、重複しているものも存在し、結果は274項目となった。漏れなく抽出することは出来たが、項目の量が多いため、絞り込みを2回実施し、チェックリストの作成を行なった。

(1) 1回目の絞り込み

シートに取り込んだ後の項目について、「ただの説明」「保守の実装前のタスク」「同じような文（同義文）」を除外した。結果、188項目に絞り込めた。絞り込んだ項目に対し、1つ1つチェック内容を設定し、チェックリストを作成した。メンバーでレビューして見た結果、チェック項目が多すぎてチェックが煩雑になるとの意見があった。従って、2回目の絞り込みを行なうことになった。

(2) 2回目の絞り込み

チェック内容の似たものを洗い出し、1回目の絞り込みと同様に統合を行なった。最終的に68項目となった。再度、メンバーでレビューして見た結果、チェック欄が yes か no かの1つではなく、暗黙知か形式知か分かるようにした方がよい、との意見があった。従って、チェック欄を2つ設けることにし、チェックリストが完成した。

3.4 チェックリストの項目

チェックリストのレイアウトとチェックリスト項目について以下に示す。チェックリストのチェック内容については、添付資料の表3-3「ソフトウェア保守実装チェックリスト」を参照のこと。

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考（理由など）
(1)	(2)	(3)	(4)	(5)	(6)

(1) プロセス実装出力

JIS 規格の「5.1.5 出力」のキーワードを載せている。複数のキーワードに関連するチェック内容については、代表的なキーワードとしている。例として、「保守の要員（要員計画を含む）、体制と役割分担」というチェック内容は、保守計画ともプロジェクト管理計画ともとれる。このようなケースの場合は、図 3-2「JIS 規格の出力イメージ」の中心のキーワードを採用することにした。

（2）CL#

チェック内容の行を特定するための 1 からの連番である。

（3）チェック内容

基本的に JIS 規格の用語をそのまま採用している。一部はチェック内容精査の過程（類似文や同義語の統合）で、JIS 規格から抽出した用語を追加し、1 つのチェック内容にしているものもある。チェック内容の文は、暗黙知か形式知か（文書化されているかいないか）を分けてチェックする目的があり、体言止めとしている。適時、語尾を読み替えてチェックすることが出来るようになっている。

例) 保守支援されるべきシステムの全体像 → システム全体像を知っている

システム全体像を俯瞰できる図表がある

例) 保守性アセスメントの指標 → 保守性アセスメントの指標を知っている

保守性アセスメントの指標が明文化されている。

（4）チェック（yes か no）（暗黙知か未知の状態か）

暗黙知になっているかのチェック欄である。「知っている」「実施している」「確立している」など、出来ている場合に yes となる。「知らない」「実施していない」「確立していない」などの場合は no となる。

（5）文書化（yes か no）（形式知になっているか）

形式知になっているかのチェック欄である。「特定されている」「最新状態を維持している」など、明文化されている場合に yes となる。「どこにあるか不明」「明文化されていても陳腐化している」などの場合は no となる。

(6) 備考（理由など）

チェック欄で no となった場合、文書化欄で no となった場合にその理由を記入する。例としては、契約外の保守作業であるため、取得者側で維持管理しているため不要、などがある。文書化欄で yes になった場合にはその文書名を記入する。

3.5 チェックリスト内容の補足説明

チェックリスト内容の用語について、補足説明を行なう。メンバーの限られた経験や知識で議論した結果による説明のため、一般的なものと比較して不足や齟齬（そご）があるかも知れない。しかし、Cグループの解釈による補足説明を加えることで、多くの人がチェックし易くなるのではないかと考え、載せることにした。

以下のカッコ付番号は CL#と対応させているため、連番にはなっていない。

(0) 取得者・提供者

JIS 規格には、「利用者」「開発者」「運用者」「保守者」「責任者」「取得者」「供給者」「第三者」「二者間」という用語がある。この用語は組織と読み替えても良いことになっている。例えば、「利用者」を利用部門、「保守者」を保守部門と読み替えても良い。これらを「保守者」と「取得者」の2つに分類した。「取得者」には、「利用者」「開発者」「運用者」「保守者」を含んでいる。メンバーは、比較的大規模なシステムを多くの組織とインターフェイスを取りながら保守を行なっている。「保守者」対「利用者」という二者間の関係だけではなく、利用者、開発者、運用者、他の保守者（営業システムと経理システムとで保守者が違うケースもあり）とのインターフェイスを取りながら保守作業を行なうことが多い。メンバーの保守現場には、ヘルプデスクやインストラクターも組織化されているところもあり、保守者がソフトウェアを直接操作して業務を行なう利用者（エンドユーザー）と、直接会話する機会は少なくなっている。

(1) システムの全体像

システムのハードウェア、ソフトウェア、およびネットワークのつながりを論理的な図表を使って表現したものである。鳥瞰図（ちょうかんず）とも俯瞰図（ふかんず）とも言われている。「保守者」と「取得者」で共有しておくことが、保守を進める上で最も重要であると考えている。システムの規模や複雑さによって、文書の形も内容も様々である。

（２）ベースライン、バージョン

これから保守を行なっていくソフトウェアの仕様、または製品のことである。保守が単一のシステムで、最新バージョンのみの維持でよければ、ベースラインの特定は簡単である。また、パッケージシステムやクライアントサーバー型システム、Web 型システムで利用者の利用環境が違う状況では、ソフトウェアのバージョン管理が必要となる。

（４）サービスレベルの規約と合意

S L A（サービスレベル合意）のことである。システム運用をアウトソーシングしている組織では I T I L が普及し、S L A を締結するところが増えてきている。ソフトウェア保守のみを行なう組織では、あまり普及していない。JIS 規格では、ソフトウェア保守においても非機能要件を含め、取得者との合意を推奨している。規約については、契約書や覚書の締結、ランク（単価テーブル）を含んでいる。

（５）費用見積り

チェックリストでは、幾つかの費用見積りを載せている。ソフトウェア保守をアウトソーシングしている場合は、数年レベルの費用見積りが必要になる。ソフトウェア保守の期間を 1 年毎など、区切って契約している場合は、その都度費用見積り（見直し）が必要になる。また、保守案件毎に契約する場合も同様である。さまざまな粒度の見積りがあるため、それぞれ分けてチェック内容とした。以下に、費用見積り項目を示す。

- ・保守を提供するための費用見積り
- ・保守性についての妥当性確認費用見積り
- ・ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用
- ・システム障害およびシステムダウン発生時の支援費用の見積り
- ・保守に必要な交通費の見積り
- ・保守者・取得者に対する教育訓練費用の見積り
- ・保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、および年間保守費
- ・保守に必要な人件費の見積り

（６）長期的なスケジュール

1 年から数年先のスケジュールのことである。ハードウェアの老朽化によるリプレース

時期、OS のサポート切れによるバージョンアップ時期、外部企業の新システム導入に伴うインターフェイス変更時期、M&A などによるビジネス環境変化に伴うシステム統合時期、法改正に伴う新ルールの適用時期などのイベントがスケジュールされる。

（7）文書管理手続き

保守にて作成される文書管理手続きのことである。文書の特定、文書の更新（作成・修正、削除）、承認、保存（維持）、閲覧（参照）の手続きである。最近では、文書管理システムと称して、文書のライフサイクルに着目したソフトウェアが導入されている保守現場もある。この場合は、文書管理手続きは確立され、標準化されていることになる。

（8）制限事項

保守作業を行なう上で、制限を受ける事柄や文書のことである。最近では、セキュリティやコンプライアンスなど、保守作業を行なうときに遵守しなければならないことが多くなった。秘密保持契約、秘密情報取り扱い規程、データセンター利用規約などの文書のことである。

（9）参照される文書

保守の実装時、保守支援されるべきソフトウェアの規模や組織が大きく、且つ長期にわたって保守を行なう場合は、保守計画の文書量が多くなる場合がある。保守計画の文書が幾つかの文書に分かれ、相互に参照している場合のことである。

（10）の1 補完文書

保守の実装時、ソフトウェア開発プロセスから引き継いだ文書、システム運用プロセスでの文書を参考にしながら、保守計画などを策定することになる。ソフトウェア開発プロセスで作成される文書の例を以下に示す。保守の多様性により文書名は例として捉えて頂きたい。

- ・保守性計画書
- ・保守計画
- ・引き継ぎ計画書
- ・移行計画
- ・品質計画書

- ・プロジェクト管理計画書
- ・構成管理計画書
- ・測定計画書
- ・検証計画書
- ・妥当性確認計画書
- ・教育訓練計画書

ソフトウェア開発プロセスやシステム運用プロセス以外にも補完文書はある。例えば、顧客からのクレーム処理は別紙「品質マニュアル」に従い対応する、と記述されていた場合は、提供者側（自社組織など）の文書が補完文書となる。

（１０）の２ 支援する文書

保守計画などを策定する中で参照されている文書である。例えば、別紙「秘密情報取り扱い規程」に従い本番データ・テストデータを取り扱う、と記述されていた場合、保守者はその文書の最新版を参照して保守作業を行なう必要がある。

（１４）妥当性確認費用見積り

大規模システムの場合、保守にて妥当性確認プロセスが工程に組み込まれる場合がある。この場合には、取得者も参画して妥当性確認が実施されるため、保守者と取得者の役割分担を明確にしておくことが必要である。また、複雑な問題や曖昧な問い合わせ、支援依頼など、想定外の保守作業が発生する可能性もあるため、十分な資源を割り当てておく必要がある。

（１８）の１ ソフトウェアの寿命

ソフトウェアベンダーが開発したソフトウェアを使い保守作業を行なっている場合、ソフトウェアベンダーのサポート期間がソフトウェアの寿命と捉えることができる。OSや言語（コンパイラ）、フレームワークのバージョンも同様である。例として、Microsoft 社の OS である Windows XP のサポート切れが、その OS 上で動いているソフトウェアの寿命である。

（１８）の２ ソフトウェアエンジニアリング環境

ソフトウェアエンジニアリング環境とは、ソフトウェア開発プロセスから引き継いだ環

境と保守を行なう中で新たに追加となった環境のことである。追加となる環境には、過去の「修正依頼および問題報告」記録やソフトウェア仕様書や設計書の変更履歴、システムやソフトウェア（運用支援含む）から出力されるログや性能データなど統計分析記録、保守を行なう過程で作成または導入したツール類などがある。

（１８）の３ ソフトウェアテスト環境

ソフトウェアテスト環境とは、主に「問題分析および修正の分析アクティビティ」と「保守レビューおよび受入れアクティビティ」にて利用する環境である。前者については、問題の特定や対応案の検討、ソフトウェア障害発生時の再現テストに利用される。後者については、ソフトウェアの検証や妥当性確認に利用される。これは、ソフトウェアの能力検証やシステム全体への影響（他のソフトウェアへの影響）が許容（想定）範囲か見極める、品質保証のための保守作業である。そのためには、本番環境に近いハードウェアとソフトウェアの環境が必要である。JIS 規格でも、ソフトウェアテスト環境を整備しておくことが推奨されている。

（１９）ソフトウェアの長期的費用

保守者がソフトウェア費用を負担している場合は、ソフトウェアの原価償却費も把握しておく必要がある。保守作業を行なうことにより、ソフトウェアの価値を維持または上げることになるため、毎年の減価償却費が変動するためである。

ソフトウェアは、自組織で開発した時、またはパッケージソフトウェアを購入した費用、カスタマイズにかかった費用は会計上、減価償却資産(無形固定資産)に該当することになる。この場合、耐用年数は５年として減価償却される。ソフトウェアをシステムに導入後、全く保守作業をしなかった場合、６年目にそのソフトウェアの資産価値はゼロになる。実際には、保守作業が行なわれるため、資産価値は６年目にゼロとはならない。因みに、供給者がパッケージソフトウェアを開発し販売した場合の耐用年数は、３年として減価償却される。

（２０）資格

公共団体などのソフトウェア保守を行なう場合、入札制度により請負先が決定されることがある。請負条件に資格が設定されている場合があるため、情報処理技術者試験、ソフトウェアベンダーの資格認定試験の合格者数や比率を維持しておく必要がある。

(29) 保守性

保守性については、ソフトウェア製品の品質—第1部：品質モデル JIS X 0129-1:2003 を参考にした。この規格では、品質特性に影響するソフトウェアの品質副特性として5つあり、測定可能な内部属性の集合により定まる、と定義されている。JIS 規格には、ソフトウェアの修正や保守作業に時間がかかるからといって保守性が悪いという意味ではない、との記述がある。定量的に状況を把握し、フィードバックして行くことで、取得者側の保守への理解が深まると考える。品質副特性と測定例を以下に示す。

- ・解析性 保守案件1件あたりの問題分析および修正分析にかかる時間
- ・変更性 保守案件1件あたりの修正の実施（開発プロセス）にかかる時間
- ・安定性 保守案件1件あたりのモジュール修正本数、テスト時のバグ件数
- ・試験性 一定規模あたりのテスト時間、障害1件あたりのテスト時間
- ・標準適合性 各規格（要領・基準・ガイドラインなど）を遵守している割合

(30) アセスメント

アセスメントについては、保守支援されるべきソフトウェアと保守の2つがある。JIS 規格では、「プロセス改善」「保守計画にフィードバック」という記述があるため、保守の方を重要視している。JIS 規格では「保守の型」について作業工数（費やす資源）を測定することを推奨している。尚、ソフトウェアについてのアセスメントは、ソフトウェア開発プロセスで重要視されている。

(31) アクティビティおよびタスク

保守の実装時、保守で扱うプロセスの全体像を示す必要はあるが、プロセスの詳細までは記述する必要はない。アクティビティとはプロセスを1つ、タスクとはアクティビティを1つブレイクダウン（ドリルダウン）したものである。JIS 規格には「作業パッケージ」という用語があり、これはWBS単位の作業であると解釈した。保守には「移行アクティビティ」がある。しかし、全てのタスクを保守者が行なうとは限らない。保守者と取得者でアクティビティおよびタスクの分担を明確にしておく必要がある。

(32) 移行要件

移行サイクル（定期的、不定期、段階的）や移行時間、障害発生時の移行などの制限事項である。取得者の移行要件を十分に把握しておく必要がある。JIS 規格では、24 時間無

停止の運用・保守サービス環境がある場合は、サービスレベルを下回ることがないように、慎重に把握する必要がある、との記述がある。切り戻しなどの作業でサービス開始時間の遅延、一度に移行したことが原因でのシステム障害を引き起こすことがないようにする。

移行要件は、ソフトウェアの切り替え、資源のリリース、イベントを想定している。従って、データ移行やシステム移行は含んでいない。

（３４）教育訓練

JIS 規格では、「保守者」「利用者」となっているが、利用者を「取得者」とした。メンバーは、比較的大規模なシステムを多くの組織とインターフェイスを取りながら保守を行なっているため、「運用者」「保守者」への教育訓練も含めることにした。

（３６）保守優先度

優先度は一般的に、システムや組織の重要度と緊急度によって決まる。保守作業は複数の案件が同時に進行している。限られた資源（保守者・期間・費用）で複数の保守案件に対応している場合は、保守案件が発生する都度、取得者と優先度を見直し、サービスレベルが低下しないよう注意が必要である。また、法令遵守など、保守者として何を大事にするかを決めておくことも重要である。

（４５）開発から引き継いだ文書

ソフトウェア開発プロセスから引き継ぐ文書の例を以下に示す。ソフトウェア保守の多様性により文書名は例として捉えて頂きたい。「（１０）の１ 補完文書」も参照のこと。

- ・ OSやミドルウェアのマニュアル（取扱説明書、文法書、メッセージ集）
- ・ 開発文書（要件定義書、仕様書、設計書）
- ・ 製造文書（文法書、プログラミング手引書、プログラマーズガイド）
- ・ テスト文書（テスト計画書、テスト仕様書、テスト手順書、テスト報告書）
- ・ 保守マニュアル（設計工程で使用するツールや記述要領）
- ・ 利用者マニュアル（操作手引き書、利用手引書、使用手引書、取扱説明書）

（４６）リポジトリ

保守作業を行なう上で必要な資源がどこにあるかを管理しているもの（ディレクトリーやメタデータベース）である。資源とは、保守支援されるべきソフトウェアそのものの以外

に、ソフトウェアエンジニアリング環境およびソフトウェアテスト環境で利用するソフトウェア（ツール類）、文書・記録・履歴情報がある。これらが特定出来ないと、保守作業の効率が低くなる。

（４７）プロセスの全体像

JIS 規格では、保守を実行するには、支援プロセスとソフトウェア開発プロセスを利用するとの記述がある。従って、それらのプロセスのどのアクティビティを利用するかを明文化する必要がある。JIS 規格に書かれている、保守のアクティビティと利用するプロセスを表 3－2 に示す。

表 3－2 「ソフトウェア保守プロセスのアクティビティと他プロセスの関係」

		ソフトウェア保守プロセスのアクティビティ					
		プロセス実装	問題分析及び修正の分析	修正の実施	保守レビュー及び受入れ	移行	破棄
他のプロセス	文書化	○	○	○		○	○
	構成管理	○				○	○
	品質保証	○	○	○	○	○	○
	共同レビュー	○		○	○	○	○
	問題解決		○				
	検証				○	○	
	妥当性確認				○	○	
	監査				○	○	○
	開発			○			

（４８）想定範囲を超えた要求

取得者から、ソフトウェアを構成するアーキテクチャの変更を伴うような要求、ソフトウェアの処理能力を超えるような要求があった場合である。その様な要求が発生した時、担当者間で揉めるのを避けるために、誰にどの様にエスカレーションするかを保守者と供給者で決定し、明文化しておくのが良い。

（４９）共同レビュー

キーワードにはヒットしなかったが、JIS 規格の「5.1.3 制御」に、『プロセス実装アクティビティの出力を制御するために、共同レビュー（JIS X 0160 の 6.6 を参照）を行なうことが望ましい。』との一文があったため、チェック内容に採用した。

(50) ソフトウェアの測定値

サービスレベル評価のためには、客観的な方法で定期的に測定できる数値が必要となる。サービスレベルに基準値が設定されていても、測定値がなければ、サービスレベルが守られているか否かの評価が出来ないことになる。

(55) アクティビティおよびタスクの明確化

保守手続きでは、アクティビティおよびタスクは詳細に明文化する必要がある。「修正依頼および問題報告」を受けた後の保守は、繰り返し行なわれることが多い。保守作業は、パレートの法則や20対80の法則に当てはまるか、それに近い比率の作業かも知れない。類似した保守が過去にあれば、保守のアクティビティおよびタスクを手順化し、テンプレートとして準備し、再利用出来るよう明文化しておく。そうしておけば、保守者による作業の漏れやダブりがなくなり、作業品質は均質化する。また、保守作業の効率も上がって行くと考ええる。JIS規格には「作業パッケージ」という用語があり、WBS単位の作業との解釈もできる。手順化はWBSをベースに行なってもよい。

(60) 保守マニュアル

保守マニュアルについては、メンバー間で「マニュアル」の定義を統一することは出来なかった。具体的名称は多岐にわたる。以下に一例を示す。いずれも不足や陳腐化していると、保守性に影響が出るものである。

- ・ソフトウェアエンジニアリング環境の利用マニュアル
- ・ソフトウェアテスト環境の利用マニュアル
- ・設計書の記述要領などの文書作成に関連するマニュアル
- ・プログラミングに関連するマニュアル
- ・テストに関連する手順書、ツールの利用手引書・使用手引書
- ・レビュー記録や報告書など保守作業に関連する記述要領などのマニュアル
- ・構成管理に関連するマニュアル（導入手引書、操作手引書、ユーザーズガイド）

(62) 引継ぎ

JIS規格では、「開発者」から新しいソフトウェアを新規で「保守者」が請負ことを想定した記述であった。保守では、「修正の実施アクティビティ」で開発プロセスを呼び出すとき、別組織に依頼するケースはあるが、新規でソフトウェア開発を請け負うことはない

め、保守組織と保守者が交替した場合とした。

（６６）利用可能性

利用可能性については、保守を行なうときの作業場所、ソフトウェアエンジニアリング環境やソフトウェアテスト環境がどれだけ利用できるかで決まる。ハードウェアの制限で、検証作業は夜間帯となるかも知れない。また、保守者と取得者が同じ組織の場合は、利用者の就業時間がサービス時間となるかも知れない。保守者がテストにて、ミドルウェアの不具合を見つけたとき、ミドルウェアの問い合わせサービス時間帯の制限を受けるかも知れない。問い合わせが出来たとしても、ミドルウェアの保守契約上の問題で受け付けされないかも知れない。このような場合は、利用可能性が低いことになる。

（６８）承認手続き

最近では、構成管理ソフトウェアやワークフローソフトウェアを導入し、ソフトウェア資産やＩＴサービスを管理する組織が増えてきた。顧客が導入したワークフローを使って保守を行なっているメンバーもいる。保守案件により、承認者が変わることもある。従って、保守を進める際には、承認手続きが明確になっている必要がある。

4. ソフトウェア保守プロセス実装チェックリストの各社評価報告

4.1 各社評価報告の概要

この章では、Cグループメンバーの保守現場にてチェックリストを使い、チェックした結果を報告する。第3章でも述べた通り、ソフトウェア保守プロセスの多様性により、Cグループメンバーの保守現場も様々である。そこで、チェックの前に、保守現場のプロフィールを載せることにした。そうすることにより、大まかな保守現場のイメージを把握することができ、チェックリストによるチェック結果の妥当性が分かったと考えたためである。

プロフィールの項目については、経済調査会の平成24年度ソフトウェア保守に関する調査票を参考にした。プロフィールの項目を表4-1「プロフィールの内容説明」に示す。

表4-1 「プロフィールの内容説明」(つづく)

No.	項目	内容
1	システム名称	基幹系（営業系、勘定系、受発注在庫など） 基幹系以外（会計、人事など）
2	保守期間	ソフトウェア保守の開始年月（西暦）
3	委託者(発注者)	府省庁、自治体、民間企業（親会社、関連会社、以外）
4	受託者(受注者)	同システムのソフトウェア開発の受託者と同一企業か否か
5	契約形態①	ソフトウェア保守のみか、システム運用も含むかなど
6	契約形態②	定額契約、実績契約、工数契約など
7	適用分野	事務系、制御系、その他
8	適用業種	製造業、情報通信業、金融業、保険業、物品賃貸業など
9	社会的影響度	殆どない（部門内）、限定的（社内）、きわめて大きい（インフラ）
10	システム構成	クライアントサーバー系、Web系、メインフレーム系
11	システムOS	Windows系、Unix系、Linux系、メインフレーム系
12	開発言語	C言語、Java、VB.NET、C#.NET、SQL、COBOL、アセンブラなど
13	データベース	Oracle、Microsoft SQL Sever、PostgreSQL、MySQLなど
14	ソフトウェア規模	プログラム本数、ソースコード行数、ファンクションポイントなど
15	保守作業割合	全体を100%とした保守の作業割合

表 4－1 「プロフィールの内容説明」（つづき）

No.	項目	内容
16	年間工数	保守の延べ作業工数
17	体制①	ソフトウェア保守延べ作業工数(年間)
18	体制②	保守者と取得者それぞれのインターフェイス
19	年間不具合発生件数	1 年間に発生した不具合数（重大、中程度、軽微に分けて）
20	課題や問題点など	ソフトウェア保守現場の課題や問題点など

4.2 各社評価報告の構成

各社評価報告の構成を以下に示す

（１）○社ソフトウェア保守概要

プロフィールの項目に従い、ソフトウェア保守の概要を説明している。説明内容が分かりづらい項目については、※１・※２などの注釈マークを付け、補足説明を行なっている。

（２）○社ソフトウェア保守プロセス実装チェック結果

チェックリストによるチェック結果である。項目が多いため、添付資料としている。

（３）○社ソフトウェア保守プロセス実装チェック評価

チェックリストのよい点・悪い点など、チェックリスト自体の評価、あるいは、チェックして気付いた保守現場の新たな課題などを報告している。

4.3 A社ソフトウェア保守のチェック結果および評価報告

(1) A社ソフトウェア保守の概要

1 システム名称	基幹系業務システム ※1
2 保守期間	2002 年 4 月～現在まで（システムは 1980 年代後半に構築）
3 委託者(発注者)	民間企業（親会社）
4 受託者(受注者)	同システムのソフトウェア開発の受託者と異なる
5 契約形態①	ソフトウェア保守サービスのみを単独で契約する ※2
6 契約形態②	定額契約（年間固定、一括請負、3 カ月契約）
7 適用分野	事務系と制御系の中間 ※1
8 適用業種	保険業
9 社会的影響度	社会的影響が限定されるシステム
10 システム構成	メインフレームシステム
11 システム OS	MSP
12 開発言語	アセンブラ、COBOL、EASY
13 データベース	ネットワーク DB
14 ソフトウェア規模	約 1,000Kstep
15 保守作業割合	是正保守 0%、予防保守 20%、適応保守 80%、完全化保守 0%
16 年間工数	48 人月（保守開発の規模が大きい案件については別契約）
17 体制①	契約先はメーカー。保守案件は顧客の情報子会社。 ※2
18 体制②	要員 4 名（協力会社 1 名） ※2
19 年間不具合発生件数	軽微なもの 1 件(2012 年 7 月～2013 年 6 月の 1 年間)
20 課題や問題点など	※3

※1 システム名称と適用分野の補足説明

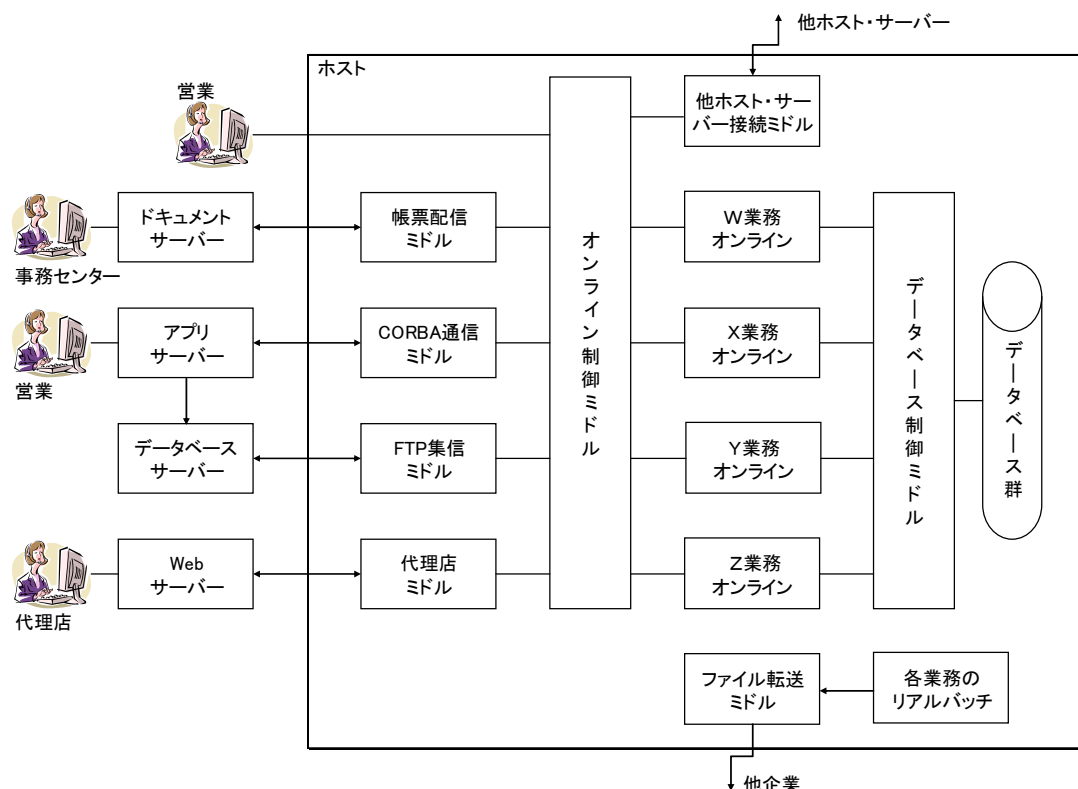
ホスト系とサーバー系のミドルウェア保守を請け負っているが、今回はホスト系ミドルウェア保守についての事例紹介を行なう。

ミドルウェアとはメーカーやソフトベンダーのパッケージを使用し、業務システムと業務システムを繋ぐ共通的な機能のことである。例えば、次に示す機能がある。

- ・業務オンラインのジョブの起動・停止・異常監視機能
- ・業務オンラインのメニュー・セキュリティ制御機能
- ・データベースのアクセス制御、
- ・他企業へのファイル転送機能(CORDEX・HULFT)
- ・他ホストやサーバーとの連携機能(Web・CORBA 通信)

予防保守・適応保守が中心であり、他に問い合わせ対応、作業依頼対応、開発運用支援(性能監視・評価など)を行なっている。是正保守、および完全化保守は行なっていない。

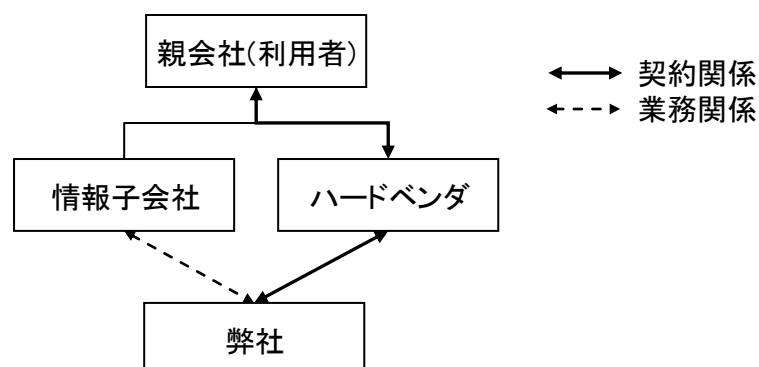
ホストミドルウェア概念図



※2 契約形態と体制の補足説明

ホスト系のソフトウェア保守業務の契約先はメーカーである。一括請負契約であり、契約期間は3ヶ月毎である。維持枠で対応出来ない大規模な保守開発案件が発生した場合は個別契約となる。保守作業については主に、情報子会社の業務システム開発部門とシステム管理部門から発生する。システム運用については、アウトソーサー企業のデータセンターにアウトソーシングしている。

契約関係と業務の関係図



※3 ソフトウェア保守の現場の課題や問題点など

保守現場では、契約先と業務依頼元が違っているため、業務量とコストのバランスを取りながら保守作業をしなければならない。従って、業務量増に伴うコスト超過のリスクを常に抱えている状態である。

（2）A社ソフトウェア保守プロセス実装チェック結果

添付資料の表4－3「A社ソフトウェア保守プロセス実装チェックリスト」参照

（3）A社ソフトウェア保守プロセス実装チェック評価

まず、チェックリストの体裁や内容評価について述べる。もう少し分類した方が良いのではないかと感じた。例えば費用見積り、様々な文書の種類の特性、組織、コミュニケーション、マネジメントなど、JIS規格の出力イメージにとらわれず、独自の切り口でチェックリストを分類すれば、より良いものになるのではないかと感じた。また、チェックリストの結果から、保守組織の強み・弱みが視覚的に分かるレーダーチャートに出来たら、もっと素晴らしいものになるのではないかと感じた。

次に保守現場の評価について述べる。情報子会社先に常駐し、顧客と一緒にオンサイトでソフトウェア保守作業を行なっているため、文書については顧客が管理しているものが多かった。この点では、特にnoと言う評価でもよいと考えている。また、文書化も余りされていない傾向が分かったが、少人数で保守作業を行なっているため、不便は感じていない。ホスト系保守技術者のローテーションはあまりない（塩漬け）ことから、問題意識が

薄いのかも知れない。

チェックの結果、S L A・測定計画・保守性アセスメント・引継ぎ計画が全く明文化されていないことに気がついた。「気付かせる」という点については、チェックリストは評価できると判断する。

チェック内容の「文書の特定」、「文書の参照」について、思い当たる事があるため、以下に紹介する。

取得者側企業の I T 全般統制の導入と合併、取得者側企業の情報子会社の合併がここ数年立て続けにあった。その都度「開発標準」や「運用ルール」など、取得者側企業で変わった。

変わったことを知らず、保守案件対応時、ドキュメントが大幅焼き直しとなったことがあった。過去のドキュメントが再利用出来なくなったのである。また、テスト工程において過去にテストで使用した本番データを参照し、厳重注意を受けたこともあった。情報子会社とのコミュニケーション不足と言えばそれまでだが、情報子会社も混乱を起こしており、どうしたら良いかお互いに分からない状況であった。最近では、取得者側でプロジェクト案件などの工程を管理するパッケージソフトウェアが導入された。

この時は、過去の経験を活かして、早急に顧客との話し合いの場を設け、レクチャーを受けた。それでも、工程管理ツールの仕組みが良く理解出来ず、工程管理上で遅延扱いを起こした。また、W B S の入力不備のため手戻りが発生したこともあった。最近では顧客から、「前回上手く行ったからと言って、今回も上手く行くとは限らない」、「過去と同じ、は通用しない」としつこく言われている。この影響により、現場では保守作業開始時に、文書の確認をよく行なうようになった。

保守現場では今後、ホスト統合のイベントが予定されており、現ホストのインターフェイスの変更（適応保守）、新ホストの構築、ソフトウェア保守業務が増大する見込みである。このチェックリストを活用して、しっかりとした保守計画立案を実施する必要があると考えている。

4.4 B社ソフトウェア保守のチェック結果および評価報告

(1) B社ソフトウェア保守の概要

1 システム名称	基幹系業務システム
2 保守期間	2007 年 6 月
3 委託者(発注者)	民間企業 (親会社・関連会社以外)
4 受託者(受注者)	ソフトウェア保守の受託者は、同システムのソフトウェア開発の受託者と同一企業
5 契約形態①	ソフトウェア保守サービスはシステム運用作業に含めて契約
6 契約形態②	工数契約
7 適用分野	事務系
8 適用業種	情報通信業
9 社会的影響度	社会的影響が極めて大きいシステム
10 システム構成	クライアントサーバシステム
11 システム OS	Unix、Linux
12 開発言語	C++、Java (Java Script、JSP)
13 データベース	Oracle
14 ソフトウェア規模	ソースコード行数 2Mstep (自社担当分)
15 保守作業割合	是正保守 10%、予防保守 5%、適応保守 10%、完全化保守 75%
16 年間工数	250 人月 (自社担当分)
17 体制①	委託者側と受託者側の双方に S E を配置し受託側は主に開発を担当
18 体制②	20 名
19 年間不具合発生件数	重大 0 件、中程度 5 件、軽微 10 件
20 課題や問題点など	※ 1

※ 1

ソフトウェアの保守工数が機能追加・仕様変更の開発工数にすべて含まれている契約のため、維持管理を行なうために必要な体制が発注者と合意しづらい。

(2) B社ソフトウェア保守プロセス実装チェック結果

添付資料の表 4-4 「B社ソフトウェア保守プロセス実装チェックリスト」参照

(3) B社ソフトウェア保守プロセス実装チェック評価

対象システムが大規模であり、かつ自社の役割が大手ベンダーによる保守の二次受注先であることから、本チェックリストで定義している「取得者」となるエンドユーザとの実際の合意事項まで見通せていないことが明らかとなった。

また、ベンダーとの契約形態がほぼ業務委託形態であり、長期にわたって保守開発を行っていることから、暗黙的な保守プロセスとなっていると言える。

しかしながら、研究テーマとしている保守プロセスの実装に関する事項を理解していることはいかなる立場においても必要な知識体系であり、特に見積りにおいては発注側と受注側でお互いに納得感のある合意形成を得るために、今後活用を図りたい。

4.5 C社ソフトウェア保守のチェック結果および評価報告

(1) C社ソフトウェア保守の概要

1 システム名称	基幹系業務システム
2 保守期間	2006 年 8 月～現在まで
3 委託者(発注者)	公益法人
4 受託者(受注者)	同システムのソフトウェア開発の受託者と同一企業
5 契約形態①	ソフトウェア保守サービスはシステム運用作業に含めて契約する
6 契約形態②	定額契約（年間固定、一括請負）
7 適用分野	事務系
8 適用業種①	公益法人
9 社会的影響度	社会的影響が限定されるシステム
10 システム構成	クライアントサーバシステム
11 システム OS	Windows 系
12 開発言語	VB.NET
13 データベース	Oracle
14 ソフトウェア規模	画面数：366、帳票数：281、ファイル数：95、バッチ数：25
15 保守作業割合	是正保守 75%、予防保守 5%、適応保守 5%、完全化保守 5%
16 年間工数	14 人月
17 体制①	委託者と直契約。基本的には顧客のシステム運用担当とやり取りを行なう。
18 体制②	要員 3 名（協力会社 2 名）1 名はプロジェクト管理のみ
19 年間不具合発生件数	中程度：2 件、軽微：3 件(2012 年 7 月～2013 年 6 月の 1 年間)
20 課題や問題点など	※ 1

※ 1

顧客の特性として毎年決済者が入れ替わるため、システム予算（維持・投資を含む）の振れ幅が激しい。特にシステム保守については、顧客の現場は業務を行なう上で必要不可欠と考えているが、決済者から費用削減の観点で、毎年必要性の是非を問われ、説明を行なっている。

また、顧客の費用感に見合うよう、C社要員 2 名に対し 2 人月以下の費用で保守サービスを提供している。そのため、保守の待機工数を他プロジェクトの作業へ振り分けるこ

とで、利益を確保できるようコントロールする必要があるが、担当する保守作業状況に加え、複数の他プロジェクトの状況まで把握する必要があり、管理に手間がかかっている。

（２）Ｃ社ソフトウェア保守プロセス実装チェック結果

添付資料の表４－５「Ｃ社ソフトウェア保守プロセス実装チェックリスト」参照

（３）Ｃ社ソフトウェア保守プロセス実装チェック評価

チェックの結果、保守実施に必要な手法は知っていることから、実運用も問題なく行なえているが、属人化しており文書化されていない、つまり引継ぎが難しい状態であると分析できる。また、特に教育・訓練や保守性アセスメントに関連した項目が出来ていない・文書化もされていないことから、長期的保守を見据えた計画を行なうのが難しい状態であると言える。

以上のように保守業務に対しチェックリストを実施し、文書化されていない項目を分析することにより、今年度のテーマでもある若手技術者への伝承が必要な知識・技術の洗い出し、ならびに引継ぎに向けた計画へと活用できると考える。

4.6 D社ソフトウェア保守のチェック結果および評価報告

(1) D社ソフトウェア保守の概要

1 システム名称	基幹系業務システム
2 保守期間	2008 年 4 月～現在まで
3 委託者(発注者)	民間企業(関連会社)
4 受託者(受注者)	ソフトウェア保守の受託者は、同システムのソフトウェア開発の受託者と異なる企業
5 契約形態①	ソフトウェア保守サービスはシステム運用作業に含めて契約している
6 契約形態②	定額＋実績契約
7 適用分野	事務系
8 適用業種①	不動産業、物品賃貸業
9 社会的影響度	社会的影響が限定されているシステム
10 システム構成	メインフレームシステム、Web 系
11 システム OS	Windows 系、Linux
12 開発言語	COBOL、Java
13 データベース	DB2、Oracle
14 ソフトウェア規模	画面数 800
15 保守作業割合	是正保守 80%、予防保守 5%、適応保守 10%、完全化保守 5%
16 年間工数	98 人月
17 体制①	委託者側は企画を担当、受託者側にて保守を実施している
18 体制②	15 名
19 年間不具合発生件数	重大 2 件、中程度 10 件、軽微 50 件
20 課題や問題点など	※ 1

※ 1

一部のシステムにおいて、ソフトウェア開発の受託者と、ソフトウェア保守の受託者が異なる。上記システムは、開発時に保守担当者が開発に参画しておらず、また開発担当者がそのまま保守担当を行わないため、該当システムにおける保守要員のスキルが低い

(2) D社ソフトウェア保守プロセス実装チェック結果

添付資料の表 4－6 「D社ソフトウェア保守プロセス実装チェックリスト」参照

（３）D社ソフトウェア保守プロセス実装チェック評価

委託者と受託者間にて取り交わされている契約書類等により、保守対象ソフトウェアは定義されているがその保守範囲を明確に定義していないため、CL#16、48、49 が yes とできなかった。保守想定範囲を超えた要求があった場合、契約に照らし合わせ保守範囲外であることを伝えられる場合と、範囲が不明瞭であるため発生都度その適用範囲について議論を行なう場合などがあるため、保守範囲を明確にした文書の作成とその合意が必要であると感じた。

保守者に必要な資格、知識も基準は無く保守品質の維持および新任技術者への伝承を進めるためにも必要資格、知識を定義し文書化する必要性を認識した。

5. おわりに

5.1 今年度の総括

Cグループは今年度、「若手保守技術者に知識や技術を早く確実に伝承するにはどうすればよいか？」をテーマとして研究に取り組んだ。全体合宿の基調講演で、標準化がソフトウェア保守の効率化に繋がることを知った。若手保守技術者に早く確実に伝承するには、標準化を進めることが良い手段であると考えた。ソフトウェア保守プロセスの最初のアクティビティである、プロセス実装アクティビティをチェックリスト化し、標準化を進める計画を立てた。

しかし、当初立てた計画より大幅に遅れを出す結果となった。チェックリストはお試し版（試供品）程度のレベルに留まり、まだまだブラッシュアップの余地が残っている状況である。また、メンバーの保守現場における標準化状況について、チェックを実施することはできたが、チェック結果を分析し標準化を進めるための施策提言には至っていない。

遅れの要因を以下に列挙する。

- ・今年度の活動計画を立てるときに、JIS 規格の認識合わせの期間を考慮していなかった。
- ・JIS 規格の中で、プロセス実装アクティビティの量が一番多いことに気付いてなかった。
- ・メンバーは保守現場でリーダーを務めているため、全員集まれる機会が一度もなかった。
- ・ベテラン研究員が2名抜けたことにより、グループのパフォーマンスが低下した。

但し、メンバーにとっては、JIS 規格の用語をよく考える機会となり、JIS 規格の理解も深まった。このような活動は、若手保守技術者への知識や技術の伝承にも役立つと考える。

5-2. 次年度に向けて

次年度については、今年度のテーマを継続する予定である。計画では、問題分析および修正の分析アクティビティに進むことになっている。今年度と同様な手法で、このアクティビティをチェックリスト化し、保守現場の標準化（文書化）を推進することで、若手保守技術者への伝承問題を解決する取り組みを行なう。

尚、メンバーからは、チェックリストをお試し版のままにしておくのはいかがなものか、と言う意見も出ている。また、今年度の統括を踏まえ、活動の進め方を見直すことも必要であると考ええる。次年度の活動案について以下に示す。

- ・今年度できなかったフォーラムを開催し、有識者の意見を反映したチェックリストにする。チェックリストのブラッシュアップを図って行く。

- ・次のアクティビティのチェックリスト化を進める。最終的に今年度作成したチェックリストとマージを行ない、ソフトウェア保守プロセス全体のチェックリストを目指して行く。

- ・JIS 規格からのチェックリストで標準化を進めるのではなく、別の解決策を考える研究に取り組む。

どの案で取り組むかについては、全体合宿の場で参加者同士の議論を行ない決定したい。次年度も継続してソフトウェア保守の問題解決の研究に取り組んで行く。

最後に、議論する場所（会議室）を提供頂いた各社には、この場をお借りして、御礼を申し上げます。

以 上

添付資料 表3-3 ソフトウェア保守プロセス実装チェックリスト（1／3）

チェック:知っている。実施している。確立している。文書化:特定している。維持している。

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	1	保守支援されるべきシステムの全体像	yes no	yes no	
保守計画	2	保守支援されるべきソフトウェアのベースライン、バージョン(関連するドキュメントを含む)	yes no	yes no	
保守計画	3	保守支援されるべきソフトウェアの保守の必要性	yes no	yes no	
保守計画	4	保守のサービス要求事項と維持すべきサービスレベルの規約と合意	yes no	yes no	
保守計画	5	保守を提供するための費用見積り	yes no	yes no	
保守計画	6	保守を提供するための長期的なスケジュール	yes no	yes no	
保守計画	7	保守計画の文書管理手続き	yes no	yes no	
保守計画	8	保守の制限事項に関する文書	yes no	yes no	
保守計画	9	保守計画で参照される文書	yes no	yes no	
保守計画	10	保守計画の補完又は保守実施を支援する文書	yes no	yes no	
保守計画	11	保守計画を理解するための用語定義	yes no	yes no	
保守計画	12	保守計画を理解するための略語及び記法	yes no	yes no	
保守計画	13	社内IT部門、外部委託、(フル)アウトソーシング、ソフトベンダー、再委託先など提供者	yes no	yes no	
保守計画	14	保守性についての妥当性確認費用見積り	yes no	yes no	
保守計画	15	保守にて維持すべき文書化の範囲とレベル	yes no	yes no	
保守計画	16	取得者への引渡しがある場合、保守者の支援範囲の取り決め	yes no	yes no	
保守計画	17	ヘルプデスクへの引渡しがある場合、保守者の支援範囲の取り決め	yes no	yes no	
保守計画	18	保守支援されるべきソフトウェアの寿命(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のソフトウェアを含む)	yes no	yes no	
保守計画	19	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用	yes no	yes no	
保守計画	20	保守を行なう時に必要な資格	yes no	yes no	
保守計画	21	保守実施時に必要な対象分野の知識	yes no	yes no	
保守計画	22	システム障害及びシステムダウン発生時の支援費用の見積り	yes no	yes no	

添付資料 表 3－3 ソフトウェア保守プロセス実装チェックリスト（2／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	23	保守に必要な交通費の見積り	yes no	yes no	
保守計画	24	保守者・取得者に対する教育訓練費用の見積り	yes no	yes no	
保守計画	25	保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、及び年間保守費	yes no	yes no	
保守計画	26	保守に必要な人件費の見積り	yes no	yes no	
保守計画	27	保守する「保守の型」	yes no	yes no	
保守計画	28	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの保守支援期間	yes no	yes no	
保守計画	29	保守性計画の策定	yes no	yes no	
保守計画	30	保守性についてのアセスメント実施	yes no	yes no	
保守計画	31	保守で扱うアクティビティ、及びタスクの定義	yes no	yes no	
保守計画	32	保守支援を行なうシステムの決定された移行要件 移行サイクル(定期的、不定期、段階的)や移行時間、障害発生時の移行などの制限	yes no	yes no	
保守計画	33	保守にて問題が発生した場合の問題解決手続き	yes no	yes no	
保守計画	34	保守者、取得者に提供する教育訓練項目とレベル	yes no	yes no	
保守計画	35	保守の保守性アセスメント実施とプロセス改善	yes no	yes no	
保守計画	36	組織の保守優先度を決定するための要素	yes no	yes no	
保守計画	37	保守のアクティビティ又はタスクの優先順位割当方法	yes no	yes no	
保守計画	38	保守のアクティビティ又はタスクへの資源割当方法	yes no	yes no	
保守計画	39	計画的な保守の見積り手法(緊急保守を除く)	yes no	yes no	
保守計画	40	保守に関連する組織と役割分担	yes no	yes no	
保守計画	41	保守の要員(要員計画を含む)、体制と役割分担	yes no	yes no	
保守計画	42	保守に必要なソフトウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するソフトウェアを含む)	yes no	yes no	
保守計画	43	保守に必要なハードウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するハードウェアを含む)	yes no	yes no	
保守計画	44	保守を行なう場所、及び施設(データセンター、オフィス、工場、ビル)の利用規約などの文書	yes no	yes no	
保守計画	45	保守にて維持すべき文書(開発から引き継いだ文書を含む)	yes no	yes no	

添付資料 表 3-3 ソフトウェア保守プロセス実装チェックリスト (3/3)

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	46	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のリポジトリ(関連するドキュメントを含む)	yes no	yes no	
保守計画	47	保守で扱うソフトウェアプロセスの全体像	yes no	yes no	
保守計画	48	保守にて想定範囲を超えた要求があった場合の対応方針	yes no	yes no	
保守計画	49	取得者と供給者との共同レビューによる合意手続き	yes no	yes no	
保守計画	50	保守作業および保守支援されるべきソフトウェアの測定値(品質、性能)	yes no	yes no	
保守計画	51	保守に必要な標準、慣行、規約、要領(要綱)などの文書	yes no	yes no	
保守計画	52	保守のリスク	yes no	yes no	
保守計画	53	保守情報の収集と記録、及び取得者への情報提供や報告の方法	yes no	yes no	
訓練計画	54	保守者、取得者に提供する教育訓練計画書	yes no	yes no	
保守手続き	55	保守開始時に利用するアクティビティ、及びタスクの明確化	yes no	yes no	
問題解決手続き	56	取得者が修正依頼・問題報告を提出する手続き	yes no	yes no	
問題解決手続き	57	保守にて問題が発生した場合の問題解決手続き	yes no	yes no	
問題解決手続き	58	取得者に支援の依頼、修正依頼又は問題報告に対する当面の回避策の提供	yes no	yes no	
測定計画	59	取得者と供給者との間で合意された保守性の要求項目とレベル(非機能要件)	yes no	yes no	
保守マニュアル	60	仕様書、プログラムの保守マニュアル、利用者マニュアル及び導入手引のような文書の更新手続き	yes no	yes no	
利用者フィードバック計画	61	取得者に支援の依頼、修正依頼又は問題報告をフィードバックする方法	yes no	yes no	
引継ぎ計画	62	保守組織や保守者が交替した場合の引継ぎ	yes no	yes no	
引継ぎ計画	63	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境、支援サービス、及びの知識や経験の引継ぎ(関連するドキュメントを含む)	yes no	yes no	
引継ぎ計画	64	未解決又は延期された問題報告及び新規要求事項、保守期間に更新する可能性がある媒体の原本の数量及び保管場所	yes no	yes no	
保守性アセスメント	65	保守性分析の結果を保守計画にフィードバックする方法	yes no	yes no	
構成管理計画	66	保守におけるソフトウェアエンジニアリング環境及びソフトウェアテスト環境の利用可能性	yes no	yes no	
構成管理計画	67	保守にて作成・更新される文書管理	yes no	yes no	
構成管理計画	68	保守にて作成・更新される文書の承認手続き	yes no	yes no	

添付資料 表4-3 A社ソフトウェア保守プロセス実装チェックリスト（1／3）

チェック: 知っている。実施している。確立している。 文書化: 特定している。維持している。

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	1	保守支援されるべきシステムの全体像	yes	yes	運用者からの提供
保守計画	2	保守支援されるべきソフトウェアのベースライン、バージョン(関連するドキュメントを含む)	yes	yes	運用者からの提供
保守計画	3	保守支援されるべきソフトウェアの保守の必要性	yes	no	必要性を文書化したものなし
保守計画	4	保守のサービス要求事項と維持すべきサービスレベルの規約と合意	yes	no	SLAは未締結のため
保守計画	5	保守を提供するための費用見積り	yes	yes	
保守計画	6	保守を提供するための長期的なスケジュール	yes	yes	
保守計画	7	保守計画の文書管理手続き	no	no	意識していない
保守計画	8	保守の制限事項に関する文書	no	no	意識していない
保守計画	9	保守計画で参照される文書	no	no	意識していない
保守計画	10	保守計画の補完又は保守実施を支援する文書	no	no	意識していない
保守計画	11	保守計画を理解するための用語定義	no	no	意識していない
保守計画	12	保守計画を理解するための略語及び記法	no	no	意識していない
保守計画	13	社内IT部門、外部委託、(フル)アウトソーシング、ソフトベンダー、再委託先など提供者	yes	yes	
保守計画	14	保守性についての妥当性確認費用見積り	yes	yes	
保守計画	15	保守にて維持すべき文書化の範囲とレベル	no	no	ある意味手戻りが発生することあり
保守計画	16	取得者への引渡しがある場合、保守者の支援範囲の取り決め	yes	yes	
保守計画	17	ヘルプデスクへの引渡しがある場合、保守者の支援範囲の取り決め	no	no	ヘルプデスクの支援は契約上なし
保守計画	18	保守支援されるべきソフトウェアの寿命(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のソフトウェアを含む)	yes	no	運用者からの提供
保守計画	19	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用	no	no	運用者まかせ
保守計画	20	保守を行なう時に必要な資格	no	no	資格の必要性なし
保守計画	21	保守実施時に必要な対象分野の知識	yes	yes	品質計画書で明文化している
保守計画	22	システム障害及びシステムダウン発生時の支援費用の見積り	no	no	一括請負のため関係なし

添付資料 表4-3 A社ソフトウェア保守プロセス実装チェックリスト（2／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	23	保守に必要な交通費の見積り	yes	yes	実施稟議で見積もっている
保守計画	24	保守者・取得者に対する教育訓練費用の見積り	no	no	一括請負のため関係なし
保守計画	25	保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、及び年間保守費	no	no	運用者からの提供
保守計画	26	保守に必要な人件費の見積り	yes	yes	
保守計画	27	保守する「保守の型」	yes	yes	品質計画書で明文化している
保守計画	28	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの保守支援期間	yes	yes	運用者のドキュメントを特定している
保守計画	29	保守性計画の策定	yes	yes	品質計画書で明文化している
保守計画	30	保守性についてのアセスメント実施	no	no	
保守計画	31	保守で扱うアクティビティ、及びタスクの定義	yes	yes	品質計画書で明文化している
保守計画	32	保守支援を行なうシステムの決定された移行要件 移行サイクル(定期的、不定期、段階的)や移行時間、障害発生時の移行などの制限	yes	yes	運用者のドキュメントを特定している
保守計画	33	保守にて問題が発生した場合の問題解決手続き	no	no	保守性のアセスメント実施はなし
保守計画	34	保守者、取得者に提供する教育訓練項目とレベル	yes	yes	品質計画書で明文化している
保守計画	35	保守の保守性アセスメント実施とプロセス改善	no	no	保守性のアセスメント実施はなし
保守計画	36	組織の保守優先度を決定するための要素	yes	yes	
保守計画	37	保守のアクティビティ又はタスクの優先順位割当方法	yes	yes	運用者との共同レビュー
保守計画	38	保守のアクティビティ又はタスクへの資源割当方法	yes	yes	
保守計画	39	計画的な保守の見積り手法(緊急保守を除く)	yes	yes	WBSをベースに作成
保守計画	40	保守に関連する組織と役割分担	yes	yes	品質計画書で明文化している
保守計画	41	保守の要員(要員計画を含む)、体制と役割分担	yes	yes	実施稟議で明文化
保守計画	42	保守に必要なソフトウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するソフトウェアを含む)	yes	no	運用者からの提供
保守計画	43	保守に必要なハードウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するハードウェアを含む)	yes	no	基盤チームからの提供
保守計画	44	保守を行なう場所、及び施設(データセンター、オフィス、工場、ビル)の利用規約などの文書	yes	no	運用者からの提供
保守計画	45	保守にて維持すべき文書(開発から引き継いだ文書を含む)	yes	yes	品質計画書で明文化している

添付資料 表4-3 A社ソフトウェア保守プロセス実装チェックリスト(3/3)

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	46	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のリポジトリ(関連するドキュメントを含む)	yes	yes	運用者との共同レビュー
保守計画	47	保守で扱うソフトウェアプロセスの全体像	yes	yes	品質計画書で明文化している
保守計画	48	保守にて想定範囲を超えた要求があった場合の対応方針	yes	yes	運用者との共同レビュー
保守計画	49	取得者と供給者との共同レビューによる合意手続き	yes	yes	運用者との共同レビュー
保守計画	50	保守作業および保守支援されるべきソフトウェアの測定値(品質、性能)	yes	yes	品質計画書で明文化している
保守計画	51	保守に必要な標準、慣行、規約、要領(要綱)などの文書	yes	no	運用者からの提供
保守計画	52	保守のリスク	no	no	意識していない
保守計画	53	保守情報の収集と記録、及び取得者への情報提供や報告の方法	yes	yes	品質計画書で明文化している
訓練計画	54	保守者、取得者に提供する教育訓練計画書	no	no	要員交替のときのみ作成しているルールなし
保守手続き	55	保守開始時に利用するアクティビティ、及びタスクの明確化	yes	yes	品質計画書で明文化している
問題解決手続き	56	取得者が修正依頼・問題報告を提出する手続き	yes	no	メールか口頭ですませている
問題解決手続き	57	保守にて問題が発生した場合の問題解決手続き	yes	no	メールか口頭ですませている
問題解決手続き	58	取得者に支援の依頼、修正依頼又は問題報告に対する当面の回避策の提供	yes	yes	保守手順書で明文化
測定計画	59	取得者と供給者との間で合意された保守性の要求項目とレベル(非機能要件)	no	no	意識していない
保守マニュアル	60	仕様書、プログラムの保守マニュアル、利用者マニュアル及び導入手引のような文書の更新手続き	no	no	パッケージ買い取り(絶版)のため必要なし
利用者フィードバック計画	61	取得者に支援の依頼、修正依頼又は問題報告をフィードバックする方法	yes	yes	保守手順書で明文化
引継ぎ計画	62	保守組織や保守者が交替した場合の引継ぎ	no	no	要員交替のときのみ作成しているルールなし
引継ぎ計画	63	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境、支援サービス、及びの知識や経験の引継ぎ(関連するドキュメントを含む)	no	no	要員交替のときのみ作成しているルールなし
引継ぎ計画	64	未解決又は延期された問題報告及び新規要求事項、保守期間に更新する可能性がある媒体の原本の数量及び保管場所	no	no	要員交替のときのみ作成しているルールなし
保守性アセスメント	65	保守性分析の結果を保守計画にフィードバックする方法	yes	yes	顧客満足度調査で改善している
構成管理計画	66	保守におけるソフトウェアエンジニアリング環境及びソフトウェアテスト環境の利用可能性	yes	no	明文化してない
構成管理計画	67	保守にて作成・更新される文書管理	no	no	野放し状態
構成管理計画	68	保守にて作成・更新される文書の承認手続き	no	no	運用者に保守案件毎に確認している

添付資料 表4-4 B社ソフトウェア保守プロセス実装チェックリスト（1／3）

チェック: 知っている。実施している。確立している。 文書化: 特定している。維持している。

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	1	保守支援されるべきシステムの全体像	yes	no	
保守計画	2	保守支援されるべきソフトウェアのベースライン、バージョン(関連するドキュメントを含む)	yes	no	
保守計画	3	保守支援されるべきソフトウェアの保守の必要性	yes	no	
保守計画	4	保守のサービス要求事項と維持すべきサービスレベルの規約と合意	yes	no	
保守計画	5	保守を提供するための費用見積り	yes	yes	
保守計画	6	保守を提供するための長期的なスケジュール	yes	no	
保守計画	7	保守計画の文書管理手続き	yes	no	
保守計画	8	保守の制限事項に関する文書	yes	no	
保守計画	9	保守計画で参照される文書	yes	no	
保守計画	10	保守計画の補完又は保守実施を支援する文書	yes	no	
保守計画	11	保守計画を理解するための用語定義	yes	yes	
保守計画	12	保守計画を理解するための略語及び記法	yes	yes	
保守計画	13	社内IT部門、外部委託、(フル)アウトソーシング、ソフトベンダー、再委託先など提供者	yes	yes	
保守計画	14	保守性についての妥当性確認費用見積り	yes	yes	
保守計画	15	保守にて維持すべき文書化の範囲とレベル	yes	yes	
保守計画	16	取得者への引渡しがある場合、保守者の支援範囲の取り決め	yes	yes	
保守計画	17	ヘルプデスクへの引渡しがある場合、保守者の支援範囲の取り決め	no	no	
保守計画	18	保守支援されるべきソフトウェアの寿命(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のソフトウェアを含む)	yes	yes	
保守計画	19	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用	yes	no	
保守計画	20	保守を行なう時に必要な資格	no	no	
保守計画	21	保守実施時に必要な対象分野の知識	no	no	
保守計画	22	システム障害及びシステムダウン発生時の支援費用の見積り	no	no	

添付資料 表4-4 B社ソフトウェア保守プロセス実装チェックリスト（2／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	23	保守に必要な交通費の見積り	no	no	
保守計画	24	保守者・取得者に対する教育訓練費用の見積り	no	no	
保守計画	25	保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、及び年間保守費	yes	no	
保守計画	26	保守に必要な人件費の見積り	yes	no	
保守計画	27	保守する「保守の型」	no	no	
保守計画	28	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの保守支援期間	yes	no	
保守計画	29	保守性計画の策定	yes	no	
保守計画	30	保守性についてのアセスメント実施	yes	no	
保守計画	31	保守で扱うアクティビティ、及びタスクの定義	yes	yes	
保守計画	32	保守支援を行なうシステムの決定された移行要件 移行サイクル(定期的、不定期、段階的)や移行時間、障害発生時の移行などの制限	yes	yes	
保守計画	33	保守にて問題が発生した場合の問題解決手続き	yes	yes	
保守計画	34	保守者、取得者に提供する教育訓練項目とレベル	no	no	
保守計画	35	保守の保守性アセスメント実施とプロセス改善	yes	yes	
保守計画	36	組織の保守優先度を決定するための要素	yes	no	
保守計画	37	保守のアクティビティ又はタスクの優先順位割当方法	no	no	
保守計画	38	保守のアクティビティ又はタスクへの資源割当方法	no	no	
保守計画	39	計画的な保守の見積り手法(緊急保守を除く)	yes	yes	
保守計画	40	保守に関連する組織と役割分担	yes	yes	
保守計画	41	保守の要員(要員計画を含む)、体制と役割分担	yes	yes	
保守計画	42	保守に必要なソフトウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するソフトウェアを含む)	yes	yes	
保守計画	43	保守に必要なハードウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するハードウェアを含む)	yes	yes	
保守計画	44	保守を行なう場所、及び施設(データセンター、オフィス、工場、ビル)の利用規約などの文書	yes	yes	
保守計画	45	保守にて維持すべき文書(開発から引き継いだ文書を含む)	yes	yes	

添付資料 表4-4 B社ソフトウェア保守プロセス実装チェックリスト（3／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	46	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のリポジトリ(関連するドキュメントを含む)	yes	yes	
保守計画	47	保守で扱うソフトウェアプロセスの全体像	yes	yes	
保守計画	48	保守にて想定範囲を超えた要求があった場合の対応方針	yes	no	
保守計画	49	取得者と供給者との共同レビューによる合意手続き	yes	yes	
保守計画	50	保守作業および保守支援されるべきソフトウェアの測定値(品質、性能)	yes	yes	
保守計画	51	保守に必要な標準、慣行、規約、要領(要綱)などの文書	yes	yes	
保守計画	52	保守のリスク	yes	yes	
保守計画	53	保守情報の収集と記録、及び取得者への情報提供や報告の方法	yes	yes	
訓練計画	54	保守者、取得者に提供する教育訓練計画書	yes	yes	
保守手続き	55	保守開始時に利用するアクティビティ、及びタスクの明確化	yes	yes	
問題解決手続き	56	取得者が修正依頼・問題報告を提出する手続き	yes	yes	
問題解決手続き	57	保守にて問題が発生した場合の問題解決手続き	yes	yes	
問題解決手続き	58	取得者に支援の依頼、修正依頼又は問題報告に対する当面の回避策の提供	yes	yes	
測定計画	59	取得者と供給者との間で合意された保守性の要求項目とレベル(非機能要件)	yes	no	
保守マニュアル	60	仕様書、プログラムの保守マニュアル、利用者マニュアル及び導入手引のような文書の更新手続き	yes	yes	
利用者フィードバック計画	61	取得者に支援の依頼、修正依頼又は問題報告をフィードバックする方法	yes	yes	
引継ぎ計画	62	保守組織や保守者が交替した場合の引継ぎ	yes	no	
引継ぎ計画	63	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境、支援サービス、及びの知識や経験の引継ぎ(関連するドキュメントを含む)	no	no	
引継ぎ計画	64	未解決又は延期された問題報告及び新規要求事項、保守期間に更新する可能性がある媒体の原本の数量及び保管場所	yes	yes	
保守性アセスメント	65	保守性分析の結果を保守計画にフィードバックする方法	no	no	
構成管理計画	66	保守におけるソフトウェアエンジニアリング環境及びソフトウェアテスト環境の利用可能性	yes	yes	
構成管理計画	67	保守にて作成・更新される文書管理	yes	yes	
構成管理計画	68	保守にて作成・更新される文書の承認手続き	yes	yes	

添付資料 表4-5 C社ソフトウェア保守プロセス実装チェックリスト（1／3）

チェック: 知っている。実施している。確立している。文書化: 特定している。維持している。

プロセス実装出力	Cl	チェック内容	チェック	文書化	備考(理由など)
保守計画	1	保守支援されるべきシステムの全体像	yes	yes	[システム関連図]
保守計画	2	保守支援されるべきソフトウェアのベースライン、バージョン(関連するドキュメントを含む)	yes	no	バージョンを明文化した文書は存在しない
保守計画	3	保守支援されるべきソフトウェアの保守の必要性	yes	no	必要性を明文化した文書は存在しない
保守計画	4	保守のサービス要求事項と維持すべきサービスレベルの規約と合意	yes	yes	【見積提案書】
保守計画	5	保守を提供するための費用見積り	yes	yes	【見積提案書】
保守計画	6	保守を提供するための長期的なスケジュール	no	no	1年単位のスケジュールしか行なっていない
保守計画	7	保守計画の文書管理手続き	yes	yes	[変更管理運用手順書]
保守計画	8	保守の制限事項に関する文書	no	no	【見積提案書】
保守計画	9	保守計画で参照される文書	yes	yes	[プロジェクト計画書]
保守計画	10	保守計画の補完又は保守実施を支援する文書	no	no	[プロジェクト計画書]
保守計画	11	保守計画を理解するための用語定義	no	no	特に用語定義は必要ない
保守計画	12	保守計画を理解するための略語及び記法	no	no	特に略語・記法は必要ない
保守計画	13	社内IT部門、外部委託、(フル)アウトソーシング、ソフトベンダー、再委託先など提供者	yes	yes	[プロジェクト計画書]
保守計画	14	保守性についての妥当性確認費用見積り	no	no	妥当性確認自体行なっていない
保守計画	15	保守にて維持すべき文書化の範囲とレベル	yes	yes	[変更管理運用手順書]
保守計画	16	取得者への引渡しがある場合、保守者の支援範囲の取り決め	yes	yes	【見積提案書】
保守計画	17	ヘルプデスクへの引渡しがある場合、保守者の支援範囲の取り決め	yes	yes	【見積提案書】
保守計画	18	保守支援されるべきソフトウェアの寿命(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のソフトウェアを含む)	yes	no	寿命を明文化した文書は存在しない
保守計画	19	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用	no	no	1年単位の費用策定しか行なっていない
保守計画	20	保守を行なう時に必要な資格	yes	no	資格を明文化した文書は存在しない
保守計画	21	保守実施時に必要な対象分野の知識	yes	no	知識を明文化した文書は存在しない
保守計画	22	システム障害及びシステムダウン発生時の支援費用の見積り	no	no	当該支援の想定をしていない

添付資料 表4-5 C社ソフトウェア保守プロセス実装チェックリスト（2／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	23	保守に必要な交通費の見積り	yes	yes	【見積提案書】
保守計画	24	保守者・取得者に対する教育訓練費用の見積り	no	no	教育訓練自体の計画がない
保守計画	25	保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、及び年間保守費	yes	yes	[予算表]
保守計画	26	保守に必要な人件費の見積り	yes	yes	[予算表]
保守計画	27	保守する「保守の型」	yes	no	型を明文化した文書は存在しない
保守計画	28	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの保守支援期間	no	no	1年単位しか期間の設定を行っていない
保守計画	29	保守性計画の策定	no	no	策定の実施がない
保守計画	30	保守性についてのアセスメント実施	no	no	アセスメントの実施がない
保守計画	31	保守で扱うアクティビティ、及びタスクの定義	yes	no	定義を明文化した文書は存在しない
保守計画	32	保守支援を行なうシステムの決定された移行要件 移行サイクル(定期的、不定期、段階的)や移行時間、障害発生時の移行などの制限	yes	no	移行要件を明文化した文書は存在しない
保守計画	33	保守にて問題が発生した場合の問題解決手続き	yes	yes	[保守運用手順書]
保守計画	34	保守者、取得者に提供する教育訓練項目とレベル	no	no	教育訓練自体の計画がない
保守計画	35	保守の保守性アセスメント実施とプロセス改善	no	no	アセスメントの実施がない
保守計画	36	組織の保守優先度を決定するための要素	yes	yes	[サービスデスク運用手順書]
保守計画	37	保守のアクティビティ又はタスクの優先順位割当方法	yes	yes	[サービスデスク運用手順書]
保守計画	38	保守のアクティビティ又はタスクへの資源割当方法	yes	yes	[サービスデスク運用手順書]
保守計画	39	計画的な保守の見積り手法(緊急保守を除く)	yes	no	手法を明文化した文書は存在しない
保守計画	40	保守に関連する組織と役割分担	yes	yes	[予算表]
保守計画	41	保守の要員(要員計画を含む)、体制と役割分担	yes	yes	[予算表]
保守計画	42	保守に必要なソフトウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するソフトウェアを含む)	yes	no	ソフトを明文化した文書は存在しない
保守計画	43	保守に必要なハードウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するハードウェアを含む)	yes	no	ハードを明文化した文書は存在しない
保守計画	44	保守を行なう場所、及び施設(データセンター、オフィス、工場、ビル)の利用規約などの文書	yes	yes	【見積提案書】
保守計画	45	保守にて維持すべき文書(開発から引き継いだ文書を含む)	yes	yes	[変更管理運用手順書]

添付資料 表4-5 C社ソフトウェア保守プロセス実装チェックリスト（3／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	46	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のリポジトリ（関連するドキュメントを含む）	yes	yes	[変更管理運用手順書]
保守計画	47	保守で扱うソフトウェアプロセスの全体像	yes	no	プロセスを明文化した文書は存在しない
保守計画	48	保守にて想定範囲を超えた要求があった場合の対応方針	no	no	当該対応の想定をしていない
保守計画	49	取得者と供給者との共同レビューによる合意手続き	yes	no	手続きを明文化した文書は存在しない
保守計画	50	保守作業および保守支援されるべきソフトウェアの測定値（品質、性能）	yes	no	測定値を明文化した文書は存在しない
保守計画	51	保守に必要な標準、慣行、規約、要領（要綱）などの文書	yes	yes	[詳細設計標準]
保守計画	52	保守のリスク	yes	no	リスクを明文化した文書は存在しない
保守計画	53	保守情報の収集と記録、及び取得者への情報提供や報告の方法	yes	yes	[保守運用手順書]
訓練計画	54	保守者、取得者に提供する教育訓練計画書	no	no	教育訓練自体の計画がない
保守手続き	55	保守開始時に利用するアクティビティ、及びタスクの明確化	yes	no	タスクを明文化した文書は存在しない
問題解決手続き	56	取得者が修正依頼・問題報告を提出する手続き	yes	yes	[サービスデスク運用手順書]
問題解決手続き	57	保守にて問題が発生した場合の問題解決手続き	yes	yes	[保守運用手順書]
問題解決手続き	58	取得者に支援の依頼、修正依頼又は問題報告に対する当面の回避策の提供	yes	yes	[保守運用手順書]
測定計画	59	取得者と供給者との間で合意された保守性の要求項目とレベル（非機能要件）	no	no	取り決め自体がない
保守マニュアル	60	仕様書、プログラムの保守マニュアル、利用者マニュアル及び導入手引のような文書の更新手続き	yes	no	手続きを明文化した文書は存在しない
利用者フィードバック計画	61	取得者に支援の依頼、修正依頼又は問題報告をフィードバックする方法	yes	yes	[サービスデスク運用手順書]
引継ぎ計画	62	保守組織や保守者が交替した場合の引継ぎ	yes	no	引継ぎを明文化した文書は存在しない
引継ぎ計画	63	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境、支援サービス、及びの知識や経験の引継ぎ（関連するドキュメントを含む）	yes	no	引継ぎを明文化した文書は存在しない
引継ぎ計画	64	未解決又は延期された問題報告及び新規要求事項、保守期間に更新する可能性がある媒体の原本の数量及び保管場所	yes	no	明文化した文書は存在しない
保守性アセスメント	65	保守性分析の結果を保守計画にフィードバックする方法	no	no	保守性分析自体がない
構成管理計画	66	保守におけるソフトウェアエンジニアリング環境及びソフトウェアテスト環境の利用可能性	no	no	利用可能性の策定自体がない
構成管理計画	67	保守にて作成・更新される文書管理	yes	yes	[変更管理運用手順書]
構成管理計画	68	保守にて作成・更新される文書の承認手続き	yes	yes	[変更管理運用手順書]

添付資料 表4-6 D社ソフトウェア保守プロセス実装チェックリスト（1／3）

チェック: 知っている。実施している。確立している。文書化: 特定している。維持している。

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	1	保守支援されるべきシステムの全体像	yes	yes	保守システム一覧
保守計画	2	保守支援されるべきソフトウェアのベースライン、バージョン(関連するドキュメントを含む)	yes	no	
保守計画	3	保守支援されるべきソフトウェアの保守の必要性	yes	no	
保守計画	4	保守のサービス要求事項と維持すべきサービスレベルの規約と合意	yes	yes	SLA合意文書
保守計画	5	保守を提供するための費用見積り	yes	yes	契約関連資料
保守計画	6	保守を提供するための長期的なスケジュール	no	no	
保守計画	7	保守計画の文書管理手続き	yes	no	
保守計画	8	保守の制限事項に関する文書	yes	yes	契約関連資料
保守計画	9	保守計画で参照される文書	yes	no	
保守計画	10	保守計画の補完又は保守実施を支援する文書	yes	yes	作業計画書
保守計画	11	保守計画を理解するための用語定義	yes	yes	各種用語集
保守計画	12	保守計画を理解するための略語及び記法	no	no	
保守計画	13	社内IT部門、外部委託、(フル)アウトソーシング、ソフトベンダー、再委託先など提供者	yes	yes	
保守計画	14	保守性についての妥当性確認費用見積り	yes	yes	契約関連資料
保守計画	15	保守にて維持すべき文書化の範囲とレベル	yes	no	
保守計画	16	取得者への引渡しがある場合、保守者の支援範囲の取り決め	no	no	
保守計画	17	ヘルプデスクへの引渡しがある場合、保守者の支援範囲の取り決め	no	no	
保守計画	18	保守支援されるべきソフトウェアの寿命(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のソフトウェアを含む)	no	no	
保守計画	19	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの長期的費用	no	no	
保守計画	20	保守を行なう時に必要な資格	no	no	
保守計画	21	保守実施時に必要な対象分野の知識	yes	no	
保守計画	22	システム障害及びシステムダウン発生時の支援費用の見積り	no	no	

添付資料 表4-6 D社ソフトウェア保守プロセス実装チェックリスト（2／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	23	保守に必要な交通費の見積り	yes	yes	
保守計画	24	保守者・取得者に対する教育訓練費用の見積り	no	no	発生都度費用の清算
保守計画	25	保守に必要なソフトウェアエンジニアリング環境、ソフトウェアテスト環境の費用見積り、及び年間保守費	yes	yes	契約関連資料
保守計画	26	保守に必要な人件費の見積り	yes	yes	保守計画書
保守計画	27	保守する「保守の型」	yes	no	
保守計画	28	ソフトウェアライフサイクルによる保守支援されるべきソフトウェアの保守支援期間	no	no	
保守計画	29	保守性計画の策定	yes	yes	保守作業計画
保守計画	30	保守性についてのアセスメント実施	no	no	
保守計画	31	保守で扱うアクティビティ、及びタスクの定義	no	no	
保守計画	32	保守支援を行なうシステムの決定された移行要件 移行サイクル(定期的、不定期、段階的)や移行時間、障害発生時の移行などの制限	yes	yes	移行計画書
保守計画	33	保守にて問題が発生した場合の問題解決手続き	yes	yes	問題管理DB
保守計画	34	保守者、取得者に提供する教育訓練項目とレベル	no	no	
保守計画	35	保守の保守性アセスメント実施とプロセス改善	yes	no	
保守計画	36	組織の保守優先度を決定するための要素	no	no	
保守計画	37	保守のアクティビティ又はタスクの優先順位割当方法	yes	no	
保守計画	38	保守のアクティビティ又はタスクへの資源割当方法	no	no	
保守計画	39	計画的な保守の見積り手法(緊急保守を除く)	yes	no	
保守計画	40	保守に関連する組織と役割分担	yes	yes	組織図
保守計画	41	保守の要員(要員計画を含む)、体制と役割分担	yes	yes	組織図
保守計画	42	保守に必要なソフトウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するソフトウェアを含む)	yes	no	
保守計画	43	保守に必要なハードウェア(ソフトウェアエンジニアリング環境、ソフトウェアテスト環境で利用するハードウェアを含む)	yes	no	
保守計画	44	保守を行なう場所、及び施設(データセンター、オフィス、工場、ビル)の利用規約などの文書	yes	yes	契約関連資料
保守計画	45	保守にて維持すべき文書(開発から引き継いだ文書を含む)	yes	yes	設計書を含む各種納品物

添付資料 表 4－6 D社ソフトウェア保守プロセス実装チェックリスト（3／3）

プロセス実装出力	CL#	チェック内容	チェック	文書化	備考(理由など)
保守計画	46	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境のリポジトリ(関連するドキュメントを含む)	yes	no	
保守計画	47	保守で扱うソフトウェアプロセスの全体像	yes	yes	サービス仕様書
保守計画	48	保守にて想定範囲を超えた要求があった場合の対応方針	no	no	
保守計画	49	取得者と供給者との共同レビューによる合意手続き	yes	no	
保守計画	50	保守作業および保守支援されるべきソフトウェアの測定値(品質、性能)	yes	no	
保守計画	51	保守に必要な標準、慣行、規約、要領(要綱)などの文書	yes	yes	保守業務における各種規約文書
保守計画	52	保守のリスク	no	no	
保守計画	53	保守情報の収集と記録、及び取得者への情報提供や報告の方法	yes	yes	月次報告
訓練計画	54	保守者、取得者に提供する教育訓練計画書	no	no	
保守手続き	55	保守開始時に利用するアクティビティ、及びタスクの明確化	no	no	
問題解決手続き	56	取得者が修正依頼・問題報告を提出する手続き	yes	yes	稟議書、申請書
問題解決手続き	57	保守にて問題が発生した場合の問題解決手続き	no	no	
問題解決手続き	58	取得者に支援の依頼、修正依頼又は問題報告に対する当面の回避策の提供	yes	no	
測定計画	59	取得者と供給者との間で合意された保守性の要求項目とレベル(非機能要件)	no	no	
保守マニュアル	60	仕様書、プログラマの保守マニュアル、利用者マニュアル及び導入手引のような文書の更新手続き	yes	yes	各種設計書の作成ガイド
利用者フィードバック計画	61	取得者に支援の依頼、修正依頼又は問題報告をフィードバックする方法	no	no	
引継ぎ計画	62	保守組織や保守者が交替した場合の引継ぎ	yes	yes	引継計画書、実施記録
引継ぎ計画	63	保守支援されるべきソフトウェア、ソフトウェアエンジニアリング環境、ソフトウェアテスト環境、支援サービス、及びの知識や経験の引継ぎ(関連するドキュメントを含む)	yes	no	引継計画書、実施記録
引継ぎ計画	64	未解決又は延期された問題報告及び新規要求事項、保守期間に更新する可能性がある媒体の原本の数量及び保管場所	no	no	
保守性アセスメント	65	保守性分析の結果を保守計画にフィードバックする方法	no	yes	契約関連資料
構成管理計画	66	保守におけるソフトウェアエンジニアリング環境及びソフトウェアテスト環境の利用可能性	yes	no	
構成管理計画	67	保守にて作成・更新される文書管理	yes	yes	設計書を含む納品ドキュメント
構成管理計画	68	保守にて作成・更新される文書の承認手続き	yes	no	

「S E R Cの考える保守とは」 活動報告

メンバー

「仕事品質」改善教室
(株) 日立ソリューションズ
(株) 日立ソリューションズ
トリプル・アイ企画
NARA コンサルティング
東芝ソリューション (株)
(株) 精密形状処理研究所
(株) NTT データ CCS
(株) バイトルヒクマ
東芝ソリューション (株)
(株) 日立ソリューションズ

大島 道夫
鈴木 勝彦
高橋 宏志
高橋 芳広
奈良 隆正
野口 大輔
長谷川 亨
馬場 辰男
弘中 茂樹
増井 和也
松本 道春

目次

0. 活動概要.....	145
1. 啓蒙・広報活動.....	147
1.1. SERC フォーラム：「消費税率変更対応。本当に大丈夫？」開催	147
1.1.1. プログラム.....	147
1.1.2. 資料	148
1.1.3. 実施報告書.....	148
1.2. ソフトウェア・シンポジウム2013 WG7：ソフトウェア保守の帰還.....	151
1.2.1. 概要	151
1.2.2. 実施報告	151
1.2.3. 発表内容	151
1.2.4. 所感（ソフトウェア・シンポジウムに参加して）	152
2. ソフトウェア保守のシラバス	153
2.1. 目的（ソフトウェア保守技術者認定試験に向けて）	153
2.2. シラバス（本文）	153
2.3. 参考文献	162
3. 個人研究.....	163
3.1. 医療プロセスはソフトウェア保守プロセス改善活動の参考にできるか？（増井和也） 163	
3.2. 入門ソフトウェア障害報告書ーソフトウェア障害報告書はこう書けー（高橋芳広） 194	

0. 活動概要

作業部会活動記録

下記は、第 22 年次 D グループ作業部会会議および研究活動の経緯を時系列に示したものである。

- ソフトウェア・メンテナンス・シンポジウム 2012

日程： 2012 年 10 月 15 日（金）

場所： 全国情報サービス産業厚生年金基金会館

テーマ：『ソフトウェアの進化』

議題： 21 年次活動報告，22 年次活動計画（案）の発表

- 第 1 回定例会

日程： 2012 年 11 月 30 日（金）～12 月 1 日（土）

場所： 葉山研修センター

議題： 22 年次活動計画

- 第 2 回定例会

日程： 2012 年 12 月 21 日（金）

場所： NARA コンサルティング

議題： ソフトウェア保守のシラバス
個人研究計画

- 第 3 回定例会

日程： 2013 年 1 月 23 日（水）

場所： NARA コンサルティング

議題： ソフトウェア保守のシラバス
SERC フォーラム検討

- SERC フォーラム

日程： 2013 年 2 月 22 日（金）

場所： 日本橋公会堂

タイトル： 消費税率変更対応。本当に大丈夫？

- 第 4 回定例会

日程： 2013 年 3 月 15 日（金）

場所： NARA コンサルティング

議題： SERC フォーラムの反省

ソフトウェア保守のシラバス

- 第5回定例会

日程： 2013年4月19日（金）

場所： NARA コンサルティング

議題： ソフトウェア保守のシラバス

- 第6回定例会

日程： 2013年5月16日（木）

場所： NARA コンサルティング

議題： ソフトウェア保守のシラバス
ソフトウェア・シンポジウムのWGについて
個人研究のフォロー

- 第7回定例会

日程： 2012年6月28日（金）

場所： 東芝ソリューション

議題： ソフトウェア保守のシラバス
個人研究のフォロー

- 第8回定例会

日程： 2012年7月8（月），10日（水）

場所： 長良川国際会議場

議題： ソフトウェア・シンポジウム 2013
WG7：ソフトウェア保守の帰還

- 第9回定例会

日程： 2012年7月26日（金）

場所： NARA コンサルティング

議題： ソフトウェア保守のシラバス
個人研究フォロー

- 第10回定例会

日程： 2012年8月30日（金）～31日（土）

場所： かんぼの宿 大洗 小会議室

議題： 活動報告書作成

（文責：高橋芳）

1. 啓蒙・広報活動

本年次の啓蒙・広報活動は、SERC フォーラムを開催し、およびソフトウェア・シンポジウム 2013 の WG7：「ソフトウェア保守の帰還」に参加した。

1.1. SERC フォーラム：「消費税率変更対応。本当に大丈夫？」開催

1.1.1. コンセプト

【テーマ】

消費税率変更対応。本当に大丈夫？

【説明】

昨年（2012 年）暮れの政権交代後、行き過ぎた円高の是正が行われ、株価も上昇に転じ、消費税率の引き上げの環境が整いつつあるが、ソフトウェアやシステムの対応の準備が滞りなく行われているのか伝わってこない。

去る 1997 年にも消費税率の引き上げが行われたが、今や 15 年が経ち、当時の資料は散逸し、当時を知る技術者も少なくなったせいか、必要な作業が見えず単に税率の定数を 5 % から 8 % に替えれば良いとの話をする人もいる。消費税率引き上げ対応の修正作業見積もりや体制確保を軽視されている現場や経営者はいないだろうか？しかしながら、15 年前の改正時には旧消費税率と新消費税率とが混在する時期（経過措置）があり、今回の改正時も同様な措置が考えられる。さらに、10 % への変更時には軽減税率の採用が検討されており、さらに大きな影響が見込まれている。

そこで、当グループでは「消費税率変更対応。本当に大丈夫？」と題して、消費税引き上げ対応作業の課題を認識するためのフォーラムを開催することにした。

今回は、原田税務会計事務所殿のご協力により税務の専門家から、消費税率変更概要および業務やシステムに与えるインパクトを説明して頂くことが出来、さらに、1997 年の消費税率変更対応を踏まえた消費税率変更の課題をソフトウェア保守の観点からの報告など、盛りだくさんの内容にすることが出来た。

1.1.2. プログラム

- ・ 13:00 - 13:30 受付
- ・ 13:30 - 13:35 開会の挨拶
- ・ 13:35 - 15:20 基調講演 「消費税の改正点と実務における重要ポイント」
原田税務会計事務所 税理士 藤森 康彦氏
- ・ 15:20 - 15:40 休息
- ・ 15:40 - 16:40 研究会報告「消費税率変更の課題 ―ソフトウェア保守の観点から―」
SERC 幹事 増井 和也氏
- ・ 16:40 - 16:50 クロージング

1.1.3. 資料

1. 講演資料参照 (Pg7)

1.1.4. 実施報告書

1. 日時：2013 年 2 月 22 日（金） 13:30～16:45

2. 会場：日本橋公会堂 第 1，第 2 洋室

3. 参加者 研究員：7 名，一般：4 名（合計：11 名）

4. 講演および質疑

4.1. 13:35 - 15:20 基調講演「消費税の改正点と実務における重要ポイント」

原田税務会計事務所 税理士 藤森 康彦氏

（質疑）

（1） Q. 長期間の工事は、6 か月前(2012 年 10 月 1 日より前)に契約すると施工後でも施工前でも大丈夫か？

A. はい。契約が 6 か月前より以前であれば、引き渡しが 4 月 1 日以降でも 5%が適用される。但し、工事進行基準が適用されている場合には、着工から 3/31 までが 5%で 4/1 以降が 8%となる。4/1 をまたがる部分は、日割り計算する。

（2） Q. 資料の P3 の図で消費税の 5%の内 4%が国で 1%が地方消費税として納税するが、使途は、地方分として 43.6%となっているが、これは 8%になった時も同じ比率になりますか？

Q. 8%になったら福祉に多くいくのではないか。

A. 比率がどうなるかわからないが、福祉の意味合いで増税しているので、変わると思われる。

（3） Q. 支払いの請求はどう表現するのか？

A. 譲渡の時期で決まる。

また、仕入れ時が 4/1 前で 5%であった場合、その後同じものが返品の際は 5%で計算する。

但し、返品ではなく、変わりの物を渡す場合、同じものならそのままだが、物が違うものを渡した場合には 8%になる。

（4） Q. 国の最終は税収入が 50%未満だが、消費税 UP だけで大丈夫なのか？

A. 景気が良くなれば、法人税も増える。但し、今後は高齢化で支出も増えるので支出も抑える必要がある。

（5） Q. 固定資産の場合、5%で購入したものと 4/1 以降で購入した場合の扱いは？

A. 簿価の方式が税込経理か税抜き経理かで異なる。

（6） Q. 固定資産等を購入した場合の消費税は売上にひも付かないが、いつ控除されるのか？

A. 控除できない部分は損金になる。仕入れ控除額となり、輸出と同じ考え。

(7) Q. 海外の消費税との関係は？

A. 免税店は国外で買っているのと同じなので、日本の消費税は該当しない。

4.2. 研究会報告「消費税率に対する取組み～あるシステムでの事例～」

SERC 研究員 弘中 茂樹氏

※弘中氏急病のため中止

4.3. 15:40 - 16:40 研究会報告 「消費税率変更の課題－ソフトウェア保守の観点から－」

SERC 幹事 増井 和也氏

(1) Q. 工事進行基準は資料 P7 の図のどれに該当するのか？(奈良)

A. 図の⑤に該当する。

(2) Q. 資料 P5 の問題分析及び修正分析は、右の設計との関連は？

A. 税率を 2 つ持つのは問題分析となり、右に行くと設計になる。

(3) Q. 資料 P4 の②と③は回るのか？

A. はい。

(4) Q. 2000 年問題の時は大きな問題が発生しなかったが、なぜか？

A. マスコミが騒いだので、みんなの準備ができたから。

(5) Q. 軽減税率は、増井さんの仕事に影響するか？

A. 軽減税率はあまり影響しないと思うが、経過措置には関係するので影響を受ける。

5. 会場の様子 (写真)



< 藤森氏 >



< 増井氏 >



<会場風景>

5. 所感

ソフトウェア技術者は疎くなりがちな消費税の税務上の扱いについて丁寧に説明して頂き、大変勉強になった。また、研究員報告では、消費税率変更に向けたソフトウェアの対応が一筋縄ではいかないことが説明された。

しかしながら、参加者が合計11名と少なかった。消費税率変更を来年に控えてのホットな話題だと考えてだけに意外であった。2000年問題の対応時には、マスコミを筆頭に大騒動が起こったが、問題の発生は比較的少なかった。今回は盛り上がりが低調なだけに、逆に問題の多発が懸念される。当研究会としても、さらなる広報・啓蒙活動が必要であるとする。

<追記>

尚、この日2番目に講演する予定であった弘中氏が3月2日に永眠されたことを付記します。故弘中氏は当研究会編集『～ISO14764 による～ソフトウェア保守開発』の著者のおひとりでもあり、社業でのソフトウェアへの関わりは言うまでもなく当研究会での活動でも先頭に立ち我々をリードし、ソフトウェア保守の発展に多大なる貢献をされました。今回も本フォーラムでの発表を最期まで気にかけていられたとの事です。故弘中氏のご冥福をお祈りいたします。(合掌)

(SERC Dグループ 高橋芳 記)

— 以上 —

1.2. ソフトウェア・シンポジウム2013 WG7：ソフトウェア保守の帰還

1.2.1. 概要

今まで、ソフトウェア保守が重要であることは一部の人の認識されていたが、十分な関心が払われていなかった。そのことは、短納期が要求されるソフトウェア開発では、将来の保守性よりも、目先のスケジュールやコストが優先されることには止むを得ないことでもあった。謂わば、ソフトウェア保守は暗黒面（Dark Side）に堕ちていたのであった。

しかしながら、昨年暮れに発生した笹子トンネル事故により、建築物をいかに保守するか、保守のしやすい構造とは等、保守に対する興味関心が高まっている。今まさにソフトウェア保守が光を取戻す好機が訪れている。

この契機を生かし、元気な保守者（組織）に戻るため、ソフトウェア保守の特徴、新技術、評価方法、キャリアパス、ビジネスモデル等々を議論し、提言をまとめる。

1.2.2. 実施報告

(1) 実施日：2013年7月8日（月）、10日（水）

(2) 場所：長良川国際会議場 大会議室

(3) 参加者

- ・ 田中 一夫 アイエックスナレッジ (SERC A グループ)
- ・ 塩谷 和範 SEA/SERC (SERC A グループ)
- ・ 馬場 辰男 NTTデータCCS (SERC D グループ)
- ・ 鈴木 勝彦 日立ソリューションズ (SERC D グループ)
- ・ 増井 和也 東芝ソリューション (SERC D グループ)
- ・ 高橋 芳広 トリプル・アイ企画 (SERC D グループ) ※順不同、敬称略

1.2.3. 発表内容

(1) ソフトウェア保守の課題

(a) ソフトウェア保守が正しく評価されていない。

- ・ 投資、教育、体制、処遇の面で、ソフトウェア保守が正しく評価されていない
- ・ 目先のプロジェクト優先で、保守を考慮しない開発が横行し、ライフサイクル・コストで評価されない
- ・ 保守の評価基準がない等の理由で、保守者自身がソフトウェア保守をアピールしていない

(b) ソフトウェア保守者のモチベーション維持が困難

- ・ 保守の不手際も開発が悪いとされ、保守の存在が認識されないと思ってしまう
- ・ 保守を意識しない開発が行われている
- ・ 2次産業（開発の下）と思われる（新しいものが良い、規模が大小による評価）
- ・ 保守の定義が組織によって違う（⇒業務ソフト、パッケージソフト）
- ・ 保守はないほうが良いと思われる
- ・ （いわゆる）市民権がソフトウェア保守にはない

(2) ソフトウェア保守の解説

- システム開発を実施し本番に入ったところから、保守作業は始まる。
- 保守作業を担当している SE 累計人月数は、開発担当者合計人月数よりも多い場合もある。
- しかしこの保守作業の計数化はほとんど行われていないし、評価基準もほとんど存在していない。情報システム産業の中でも不思議な世界であるし、「紺屋の白袴」といわれても仕方がない項目のひとつである。
- 開発はひと時であるが、保守期間は半永久である。保守作業が 20 年以上にわたって継続するプロジェクトもある。20 年以上ひとつのシステムを担当し続ける人は珍しいので引き継ぎ作業が発生するが、一回引き継ぐたびにノウハウは流出し、担当者の理解は浅いものになってゆく。ドキュメントを必ず更新し、常にプログラムシートと設計仕様が一致しているシステムはむしろ珍しい。

「ソフトウェア開発管理基準に関する調査報告書（ソフトウェアメトリックス調査）」[1]より抜粋

(3) 帰還に向けて

- 多様性の考慮（業務／パッケージ、規模等.）
- ソフトウェア保守の再構築（モデル、プロセス、ツール等）

（ソフトウェア保守の価値は機会損失で計る、レジリエンスの概念を導入、等を踏まえる）

(b) 保守総合サービス

- 組織、戦略等（ex. 運用と組み合わせる）

(c) 保守担当者の教育体系の確立

(d) 保守を意識した開発の推進（ex. ソフトウェア開発時の保守性観点テスト）

1.2.4. 所感（ソフトウェア・シンポジウムに参加して）

ソフトウェア保守を世に訴える機会が少ない。S E R C 内部だけでは影響範囲があまりにも少ないとの危機感からソフトウェア・シンポジウムのWGに参加した。

参加者はS E R Cのメンバーのみであり、期待外れの観は否めないがこれが正直な実情であろう。しかしながら、WGの成果発表会の時間には「自分がいままでしていたことは保守だったのか？（驚き）」「保守は帰還できそうですか？」との質問があり。少なからず反響はあり、保守者自身のアピールと言う面で参加した効果はあったと思う。

今回のまとめた対策案の具体化を進めると共に、ソフトウェア保守アピールの維持するためにソフトウェア・シンポジウムの参加を継続し、毎年恒例にしたいと考える。

参考文献

[1] ソフトウェア開発管理基準に関する調査報告書（ソフトウェアメトリックス調査）、社団法人 日本情報システム・ユーザー協会、2012.02

（文責：高橋芳）

2. ソフトウェア保守のシラバス

2.1. 目的（ソフトウェア保守技術者認定試験に向けて）

今までのS E R C Dグループの活動ではソフトウェア保守者に必要な教育の体系が無いために、評価が曖昧になってしまう、と言う問題が摘出されている。そこで、当グループでは教育体系を確立し、技術力の評価を行うためS E R Cによるソフトウェア保守技術者の認定試験の実施を目指すことにした。その第一歩としてソフトウェア保守技術のシラバス（第1版）を作成した。これを叩き台として妥当性・網羅性はどうか等の意見を広く問うことにより磨き上げ、一般に普及したいと考える。

2.2. シラバス（本文）

1. ソフトウェア保守の基本事項

1. 1 定義と用号

●学習目標

- ・ISO/IEC/IEEE14764（以下 ISO14764）に定義された用語を理解する。
- ・ソフトウェア保守の目的を理解する。

●学習の対象となる用語、概念

ISO14764, JIS X0161 適応保守, 是正保守, 緊急保守, 保守性, 改良保守, 修正依頼, 完全化保守, 予防保守, 問題報告, ソフトウェア保守, ソフトウェア引継ぎ

1. 2 保守の性質

●学習目標

- ・ソフトウェア保守がライフサイクルを通してソフトウェア製品を支えることを理解する。
- ・保守者という述語が保守アクティビティを実行する個人を指す場合と、組織を指す場合とがあることを理解する。
- ・ISO14764 で述べられているソフトウェア保守の主要アクティビティを理解する。
- ・保守者からみた開発者との関係を理解する。

●学習の対象となる用語、概念

ソフトウェアライフサイクル, 保守プロセス, 保守者, 開発者との連携

1. 3 保守の必要性

●学習目標

- ・保守が、ソフトウェアが利用者の要求を継続して満たすために必要とされることを理解する。
- ・保守は開発時に使われたライフサイクルモデルにかかわらず必要とされることを理解する。
- ・保守の目的は、障害の是正, 設計の改善, エンハンスの実施, 他のソフトウェアとのインタフェースの維持, 別のハード・ソフト・システム要素・通信機能を使えるようにプログ

ラムを適合させること，レガシーソフトウェアの移行，ソフトウェアの廃棄の実施であることを理解する。

- ・保守者のアクティビティを構成する次の5つの重要特性を理解する。
 - 1) ソフトウェアの日々の動きの管理を維持する
 - 2) ソフトウェア修正の管理を維持する
 - 3) 既存機能を完全なものにする
 - 4) セキュリティ上の脅威を認識し，セキュリティ上の脆弱性を除去する
 - 5) ソフトウェアの性能が，容認できないレベルまで劣化することを防止する

●学習の対象となる用語，概念

利用者要求，障害の訂正，設計の改善，エンハンスの実施，他ソフトウェアとのインタフェース，動作環境の変化へのプログラム適合，レガシーソフトウェアの移行，ソフトウェアの廃棄

1. 4 保守コストの構造

●学習目標

- ・ソフトウェア保守がソフトウェアライフサイクルにかかわるコストの大部分を占めることを理解する。
- ・ソフトウェア保守コストの大部分は訂正以外のものに使われることを理解する。
- ・ソフトウェア保守コストの多くが訂正に使われているというのは勘違いであり，それは改良と訂正を区別しない管理が原因であることを理解する。
- ・ソフトウェアの保守コストは，ソフトウェアの運用およびハードウェア環境に関連することを理解する。
- ・ソフトウェア保守コストが組織の理念，競合相手，保守プロセス，製品，要因等の組織環境と関係することを理解する。

●学習の対象となる用語，概念

ソフトウェア保守コストの構造，改良と訂正，非是正アクション

1. 5 ソフトウェアの進化

●学習目標

- ・レーマンの8つの進化の法則を理解する。
- ・ソフトウェアは進化するにつれ，なんらかのアクションをとらないと複雑さが増加することを理解する。

●学習の対象となる用語，概念

レーマンの進化の法則，保守は継続的な開発

1. 6 保守の分類

●学習目標

- ・ソフトウェア保守の4分類を、訂正、改良およびプロアクティブ、リアクティブの4象限の分類で理解する。
- ・是正保守の特徴を理解する。
- ・適応保守の特徴を理解する。
- ・完全化保守の特徴を理解する。
- ・予防保守の特徴を理解する。

●学習の対象となる用語、概念

是正保守、適応保守、完全化保守、予防保守、訂正、改良、プロアクティブ（事前実施）、リアクティブ（事後実施）

2. ソフトウェア保守の主要課題

2. 1 技術的課題

2. 1. 1 限られた理解

●学習目標

- ・ソフトウェア製品の保守は、ソフトウェア製品の理解が大半であることを理解する。
- ・保守母体を完全に理解できることは稀であり、限られた理解の元で修正箇所の洗い出し、修正方法を決めるか理解する。
- ・ソフトウェア・エンジニアにとって、ソフトウェア母体の理解は、重要なことであると理解する。

●学習の対象となる用語、概念

製品理解手法（ソースコード解析、仕様書解析、マニュアル、ブラックボックステスト）、変更点管理、ソフトウェア構成管理、コスト見積もり

2. 1. 2 テスト

●学習目標

- ・ソフトウェア製品の保守後のテストの重要性を理解する。
- ・修正箇所のテストだけでなく、回帰テストの必要性を理解する。
- ・修正後のテスト環境は、時間・環境などが十分に確保できないことを理解する。

●学習の対象となる用語、概念

再現テスト、各種テスト手法理解（カバレッジテスト、外部仕様テスト、単体・結合テスト、ブラックボックステスト、回帰テストなど）、テストの自動化、コスト見積もり、計測（品質、性能）

2. 1. 3 影響分析

●学習目標

影響分析は、既存のソフトウェアの変更の影響について完全な分析をどのように行うかの記述が必要であることを理解する。

- ・保守者は、ソフトウェアの構造と内容の詳細な知識が必要なことを理解する。
- ・保守者は、ソフトウェアの変更に対しての見積もりを実施することを理解する。
- ・変更に対しての影響分析やリスクも必要なことを理解する。
- ・ISO14764 の影響分析の各タスクを理解する。
- ・問題の難しさによって、どのように修正するかが決められることがあることを理解する。
- ・保守性を念頭において設計されたソフトウェアは、影響分析が極めて容易になることを理解する。

●学習の対象となる用語、概念

影響分析、ソフトウェアの構造、変更要求(modification request)、問題報告(problem report)、ISO14764

2. 1. 4 保守性

●学習目標

- ・ソフトウェア製品の保守の機能と変更の定義を理解する。
- ・ソフトウェア開発活動中に保守コストを削減するためのコントロールの必要性を理解する。
- ・体系的、成熟したプロセス、テクニック、およびツールの存在が、ソフトウェアの保守性を高めるのに役立つことを理解する。

●学習の対象となる用語、概念

ISO14764 (2 c3s4) , 保守の機能、変更の定義、機能仕様、品質特性、ソフトウェアのマニュアル整備、体系的、成熟したプロセス

2. 2 管理上の課題

2. 2. 1 組織目標との整合性

●学習目標

- ・組織の目標とプロジェクトの目標の整合性が大切であることを理解する。
- ・上級管理職レベルでのビューが投資収益率に影響することを理解する。

●学習の対象となる用語、概念

組織の目標、保守活動への投資とリターン、プロジェクト予算、上級管理職レベルでのビュー、投資収益率

2. 2. 2 人員確保

●学習目標

- ・スタッフが保守要員を集める方法を理解する。
- ・このために保守要員の確保には、モチベーション付けが大切であることを理解する。

●学習の対象となる用語，概念

保守要員の確保と維持

2. 2. 3 プロセス

●学習目標

- ・ソフトウェアライフサイクルプロセスの概念を理解し，ソフトウェア開発アクティビティと比較することによりソフトウェア保守アクティビティを学ぶ。このアクティビティが管理の課題となる点を理解する。

●学習の対象となる用語，概念

ソフトウェアライフサイクルプロセス，ソフトウェア保守アクティビティ

2. 2. 4 保守の組織的側面

●学習目標

ソフトウェア保守の責務を担当する組織，あるいは職務をどのようにして決めるかを学ぶ。

- ・専門化を可能にする
- ・コミュニケーション経路を作る
- ・エゴレス(非利己)で，学究的な雰囲気の促進
- ・個人に対する依存（属人化）を減らす
- ・定期監査チェックを可能にする

●学習の対象となる用語，概念

ソフトウェア保守の責務を担当させる組織，恒久的な保守チームの利点，割り当てと委任

2. 2. 5 外部・海外調達

●学習目標

外部調達と海外でのソフトウェア保守は，IT産業の主要な構成要素となっている。ソフトウェア保守を含めて，ソフトウェアの作業全体を外部調達しつつあることを学ぶ。

●学習の対象となる用語，概念

要求される保守サービスの範囲，サービスレベル合意の項目，契約書の詳細，遠隔サイトのヘルプデスクは母国語の話し手を配置，有意義な初期投資，自動化を要求する保守プロセスの確立

2. 3 保守コストの見積もり

2. 3. 1 コスト見積もり

●学習目標

- ・保守のコスト見積りの基本概念を理解する。
- ・保守のコスト見積りの要点と開発のコスト見積りの要点に異なる部分があることを理解する。

●学習の対象となる用語、概念

保守コスト，影響分析，影響範囲の識別，変更に要するリソース，技術的コスト要因，非技術的コスト要因，ISO14764（7.4.1）

2. 3. 2 パラメトリックモデル

●学習目標

- ・保守要件は対応工程ごとの作業比率（負荷山積み）の形態がさまざまであることを理解する。
- ・保守のコスト見積りにおいて、適切かつ詳細に前提条件を置くことが重要であることを理解する。

●学習の対象となる用語、概念

パラメトリックモデル，数学的モデル，過去の時系列データ計測，コストドライバ（コストを大きく変化させる）属性

2. 3. 3 経験値

●学習目標

- ・保守要件は比較的短期間で対応が完了することが多いため，過去のコスト見積りとコスト実績の比較データ（経験値）が十分な数記録されていることを理解する。
- ・新たな保守コスト見積りを実施する場合，それらの経験値を参考にして見積もることで精度の高い見積りができることを理解する。

●学習の対象となる用語、概念

経験値，保守経験，保守見積もり過去データ，計測プログラム，人・時間

2. 3. 4 開発コスト見積りとの相違点

●学習目標

- ・開発コストの見積り手法をそのまま適用するとは，保守コスト見積りを精度高く見積もるには有効でないことを理解する。
- ・工程別見積り，ファンクションポイント法などを保守コスト見積りに応用する場合，開発コスト見積りにない注意点があることを理解する。

●学習の対象となる用語、概念

ひとこぶラクダモデル，ふたこぶラクダモデル，原因調査，インパクト分析，修正設計，テスト環境構築，回帰テスト

2. 4 ソフトウェア保守計測

●学習目標

- ・保守の軽量尺度は，その組織の状況に基づいてふさわしいものが決定されなければならないことを理解する。

●学習の対象となる用語，概念

解析容易性，変更容易性，安定性，テスト容易性，ソフトウェアの規模，ソフトウェアの複雑度，理解性，保守性

3. 保守プロセス

3. 1 保守プロセス

●学習目標

- ・ソフトウェア製品の保守の機能と変更の定義を理解する。
- ・ソフトウェア開発活動中に保守コストを削減するためのコントロールの必要性を理解する。
- ・体系的プロセス，成熟したプロセス，技術，およびツールの存在が，ソフトウェアの保守性を高めるのに役立つことを理解する。

●学習の対象となる用語，概念

ISO14764 (2 c3s4) ，保守の機能，変更の定義，機能仕様，品質特性，ソフトウェアのマニュアル整備，体系的，成熟したプロセス

3. 2 保守アクティビティ

3. 2. 1 特有のアクティビティ

●学習目標

- ・組織の目標とプロジェクトの目標の整合性が大切であることを理解する。
- ・上級管理職レベルによるレビューが，投資収益率に影響することを理解する。

●学習の対象となる用語，概念

組織の目標，プロジェクトの目標，保守活動への投資とリターン，プロジェクト予算，上級管理職レベルでのビュー，投資収益率

3. 2. 2 支援アクティビティ

●学習目標

- ・保守要員の確保には，モチベーション付けが大切であることを理解する。

●学習の対象となる用語，概念

保守要員の確保とモチベーション維持

3. 2. 3 保守計画アクティビティ

●学習目標

- ・保守者は計画の見地と結び付けられる多くの課題に取り組まなければならないことを説明出来る。
- ・さらにソフトウェア保守計画は新しいソフトウェア開発するという決定から始まり、コンセプト文章がメンテナンス計画に続いて作られるべきであることを理解する。

●学習の対象となる用語、概念

事業計画，保守計画，リリース/バージョン計画，個別のソフトウェア変更要求計画，保守コンセプトの内容

3. 2. 4 ソフトウェア構成管

●学習目標

- ・ソフトウェア構成管理 (SCM) の手続きに要求されるそれぞれのステップについて規定される事項を理解する。
- ・さらにソフトウェア保守の SCM は変更ソフトウェア開発のための SCM と小さな点で数多く違っていることを理解する。

●学習の対象となる用語、概念

ソフトウェア製品の識別，認可，実装，リリース，ステップの検証，妥当性確認，識別

3. 2. 5 ソフトウェア品質管理

●学習目標

- ・望ましい品質を達成するために，ソフトウェア品質保証 (SQA) のためのアクティビティと技法，つまり V&V, レビューと監査は全ての他のプロセス連携しなければならないことを理解する。

●学習の対象となる用語、概念

ソフトウェア保守の品質プログラム，開発プロセス，技法，配布可能物，テスト結果の改善

4. 保守のために技法

4. 1 プログラム理解

●学習目標

- ・プログラム変更に際して，プログラムソースコードを読み込み，理解することを学ぶ。
- ・そのために重要なのが，ソースコードを分類（整理）したり，表示したりするツールであることを理解すること。
- ・明確で簡潔なドキュメントもプログラムの理解を助けることを理解する。

●学習の対象となる用語、概念

オープンソース・ソフトウェア・コード・ブラウザ

4. 2 リエンジニアリング

●学習目標

- ・リエンジニアリングとリファクタリングを学ぶ。
- ・リエンジニアリングは、老化したレガシーなソフトウェアを新しい形式に再構成するために行う調査と改造、そしてそれに続けて実施する新しい形式の実装までを指す。リファクタリングは、振る舞いを変更せずに、プログラムを再構成することを目的としたリエンジニアリング技術を指す。
- ・リファクタリングは、プログラムの構造とその保守性の改善を目指して行われ、マイナーチェンジの中で、その技術が利用できることを理解する。

●学習の対象となる用語、概念

リエンジニアリング、レガシーソフトウェア、リファクタリング

4. 3 リバースエンジニアリング

●学習目標

- ・リバースエンジニアリングを学ぶ。
- ・リバースエンジニアリングが、ソフトウェアのコンポーネントおよびそれらの相互関係を識別し、別の形式やより高い抽象概念でのレベルのソフトウェア表現するために行う、ソフトウェアを分析するプロセスであることを理解する。
- ・リバースエンジニアリングは、ソフトウェアを変更することなく、新しいソフトウェアを生成するものでもないことを理解する。
- ・リバースエンジニアリングによって、ソースコードからコールグラフおよび制御フローを作成し、再文書化や設計情報の回復に利用されることを学ぶ。
- ・物理的データベースから論理スキーマを回復しようとするデータ・リバースエンジニアリングは、ここ数年重要性が増してきたことを学ぶ。

●学習の対象となる用語、概念

リバースエンジニアリング、コールグラフ、制御フロー、再文書化、設計情報の回復、データ・リバースエンジニアリング

4. 4 移行

●学習目標

- ・ソフトウェア保守におけるアクティビティの一つである「移行」について、保守者が実施すべき項目と、移行プロセスを理解する。

●学習の対象となる用語、概念

目的の通知、移行環境（データ形式、OS、MW等）、移行計画、移行対象（データやアプリケーション）、移行手順書、移行ツール（データ変換等）、完成の通知、新環境での評価、データの保管

4. 5 廃棄

●学習目標

- ・ソフトウェアライフサイクルにおける最終段階であるシステムの廃棄および機密保護について理解する。

●学習の対象となる用語，概念

廃棄計画，廃棄の分析，データの完全消去，セキュリティ保護，廃棄記録の保管，廃棄プロセス

5. ソフトウェア保守ツール

●学習目標

- ・ソフトウェア保守に利用するツールの種類や目的を理解する。

●学習の対象となる用語，概念

プログラム保守ツール（プログラムスライサー，静的アナライザ，動的分析アナライザ，クロスリファレンス，従属分析（依存性分析）等），リバースエンジニアリングツール，ソフトウェアのテスト，構成管理，文書化，測定ツール

2.3. 参考文献

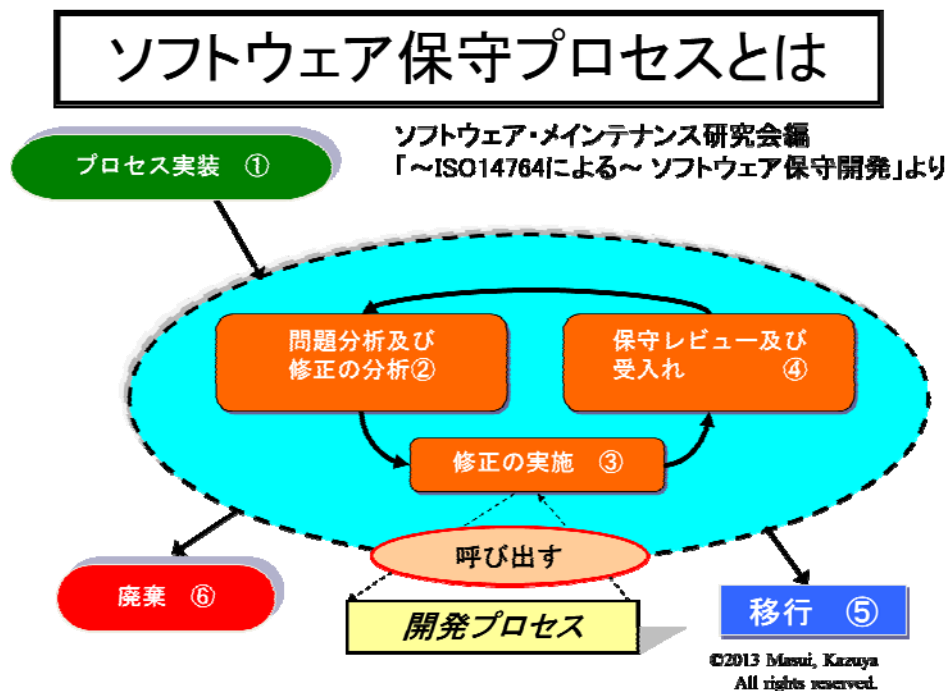
- ・ISO/IEC14764, IEEE Std 14764-2006, Software Engineering – Software Life Cycle Processes – Maintenance
- ・JIS X 0161 : 2008, ソフトウェア技術—ソフトウェアライフサイクル—保守
- ・～ISO14764 による～ソフトウェア保守開発，増井和也他著，株式会社ソフトウェアリサーチセンター発行，ISBN978-4-8873-249-4
- ・ソフトウェアエンジニアリング基礎知識体系—SWEBOK2004，松本吉弘訳，オーム社発行，ISBN-13: 978-4274500299
- ・ソフトウェア品質知識体系ガイド—SQuBOK Guide, SQuBOK 策定部会編，オーム社発行，ISBN-13: 978-4274501623
- ・共通フレーム2013 ～経営者，業務部門とともに取組む「使える」システムの実現～，独立行政法人情報処理推進機構（IPA）技術本部 ソフトウェア・エンジニアリング・センター（SEC）編集・発行，ISBN-13 : 978-4-905318-19-4

3. 個人研究

3.1. 医療プロセスはソフトウェア保守プロセス改善活動の参考にできるか？（増井和也）

はじめに

論者がソフトウェア保守の実務を担当し始めてから 35 年以上経過した。その間、さまざまなタイプの保守をさまざまな立場で担当し、どれだけ多くの既存ソフトウェアを自ら修正または修正の指示をしてきたか分からない（今も保守者として保守実務を主担当中）。加えて、現在に至るまで 25 年以上ソフトウェア保守に関する研究に係ってきた[1]。



そのような中で、5 年以上前からソフトウェア保守のプロセス[2]と医療関係者が実際に行っているプロセス（以下、単に医療プロセスと呼ぶ）との間には、対象が異なってはいても共通の考え方が多く存在するようだとはっきりと感じるようになってきた。前年度の本研究会の論者個人研究報告では、その認識をベースに、ソフトウェアをヒトや生き物にたとえ、ソフトウェアのライフサイクル（生涯）における開発と保守の位置づけを考察している[3]。

仮にソフトウェアの保守者が医療プロセス実行上の困難さ、要求される高度なスキルや高い倫理観等について共通点があるならば、保守者への評価が見直されてしかるべきと論者は考えている。またそのような発想を保守者自身が持つことで、自らのモチベーションやモラルも格段に向上することが期待したい。

本論では，ソフトウェア保守プロセス（以下，単に保守プロセスと呼ぶ）と医療プロセスの共通点を整理し，保守プロセスとして不足しがちな部分（改善の余地がある部分）を明らかにするとともに，法制度的な改善，経営層が考えるべき改善，保守者自身（個人，組織）が考えるべき改善について言及したい。

本論は，次の構成となっている。

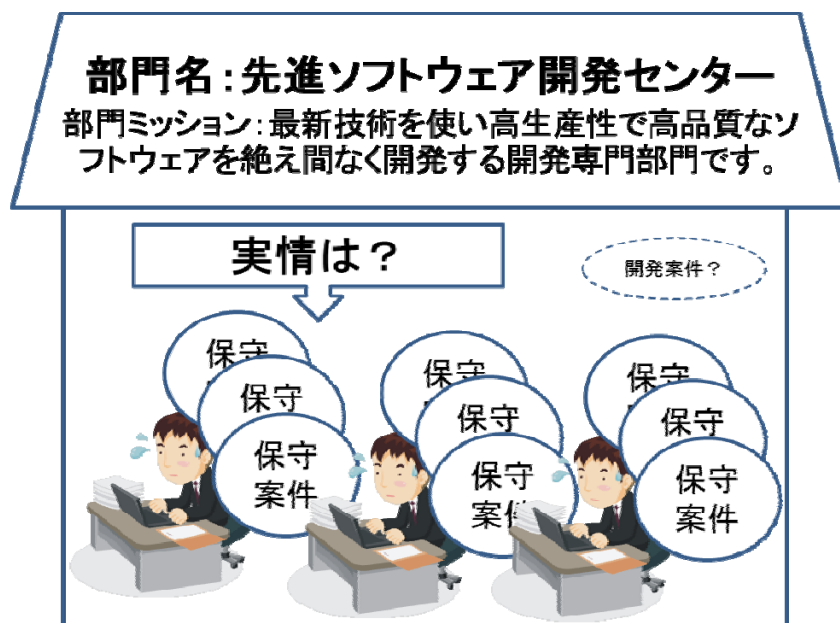
- 1．本論の目的
- 2．本論のアプローチ
- 3．保守プロセスと医療プロセスの類似点
- 4．保守プロセスの課題解決に有効な考え方が医療プロセスにあるか？
- 5．保守プロセス改善へのアプローチ
- 6．まとめ

1. 本論の目的

(1) ソフトウェアの開発は本当に主要な仕事か？

ソフトウェアの保守作業は、ユーザ企業の情報システム部門やその情報子会社、組込ソフトウェアの作成を行う会社や部門、ITベンダーのソフトウェア担当部門、SaaSやクラウドなどのサービスプロバイダのIT部門などで行われていると考える[4]。ただ、その会社や部門の名称には「開発」という言葉が使用されるとはあっても、「保守」という名称が使用されることはまれである。そうなる理由が個別に種々あるとしても、共通的には「保守」は「開発」を主として行う部門が副次的に行う作業というイメージが一般的であるためと考える。

しかし、「保守」が「開発」の副次的な作業というイメージとは逆に、その部門や会社が担当する作業の大半が「保守」であることも珍しくない。数十年前のように既存ソフトウェアがほとんどなく新規開発が主体であった時代ではもうなくなっているかもしれない。そして、社会が必要とするソフトウェアはすべてではないにせよ、大半はそろっているのが今の時代である。新規に開発する作業より、既存ソフトウェアの修正が主な作業になる傾向が無視できなくなっている。



(2) ソフトウェア技術者は無意味なギャップに悩んでいる

このように作業実態と部門の名称が乖離した部門に配属された技術者は、結局次のような思いのどれかを抱きながら大半の業務である「保守作業」をしていることはないだろうか（論者の推測）。

「配属されたらソフトウェアの新規開発をたくさん経験できると期待したのに、やっていることは10年以上前に開発された古いシステムの保守ばかり」

「最新技術の開発を行っている」と部門はPRをしているが、優秀な先輩の大半は古いシステムのソフトウェア保守ばかりをやっている」

「現行システムのリプレースの話はあるようだが、一向にリプレースへ動く気配がなく、現行

ソフトウェアの保守の終わりが見えない」

「ここ数年ずっと古いソフトウェアの保守ばかりの経験しかない。急に最新ソフトウェア技術で開発プロジェクトに行けと言われても、新人に負けてしまうかもしれない。実は自分の技術がどこまで潰しが効くのか全然自信がなくなっている」

「親会社の業績が良かった以前は大きな新規機能の追加を担当し、保守はすべて外注に任せていた。しかし、最近業績が悪くなると新規開発もなくなり、保守は社員でやれと言われるが、急に外注が保守してきた既存ソフトウェアの保守はできるわけないよ」

「組込先の装置は新しくなっても組み込むソフトは他人の作った既存ソフトの使い回しや追加・修正ばかり。これがあるべきソフトウェアの開発と呼ぶのかな？」

「既存システムをベースとした修正や改造ばかりで、ソフトウェアの新しいアルゴリズムや独創的な処理設計を考える必要はなく、大学や大学院等で学んできた最新技術がまったく役に立たない」

「ソフトウェア障害の原因調査を大至急で依頼され、原因が判明した後は決められたルーチンで修正を行い、後は同じようなテストばかり夜遅くまで繰り返している。これが本当のソフトウェア技術者の姿か？」

「この部門の経験ではITスキルを高めることはできない。プログラムの修正テストはだれかに振って、早く管理部門へ配置転換されることを目指そう」

「今の仕事はITエンジニアとして創造的な仕事としてあまり魅力を感じない」

「この職場は、学校の後輩に勧めたくない」

このように感じるソフトウェア系IT技術者が仮に多いとするとこの職業は魅力的な職業とはとてもいえないだろう。魅力的でなくなっている原因は、あるべき姿（開発中心のイメージ）と現実（保守中心の作業実態）のギャップが大きな要因の一つであると論者は考える。

(3)ソフトウェア業務を統括する人々は今も開発中心のイメージに固執していないか？

多くのソフトウェア関係者は、このギャップに対し、保守が多いのは開発技術が未熟だからと考えているかもしれない。開発技術が理想の姿に進化すれば、やがて最新の開発を中心とした理想のソフトウェア技術者の仕事が訪れるだろうとその到来を期待してきた。

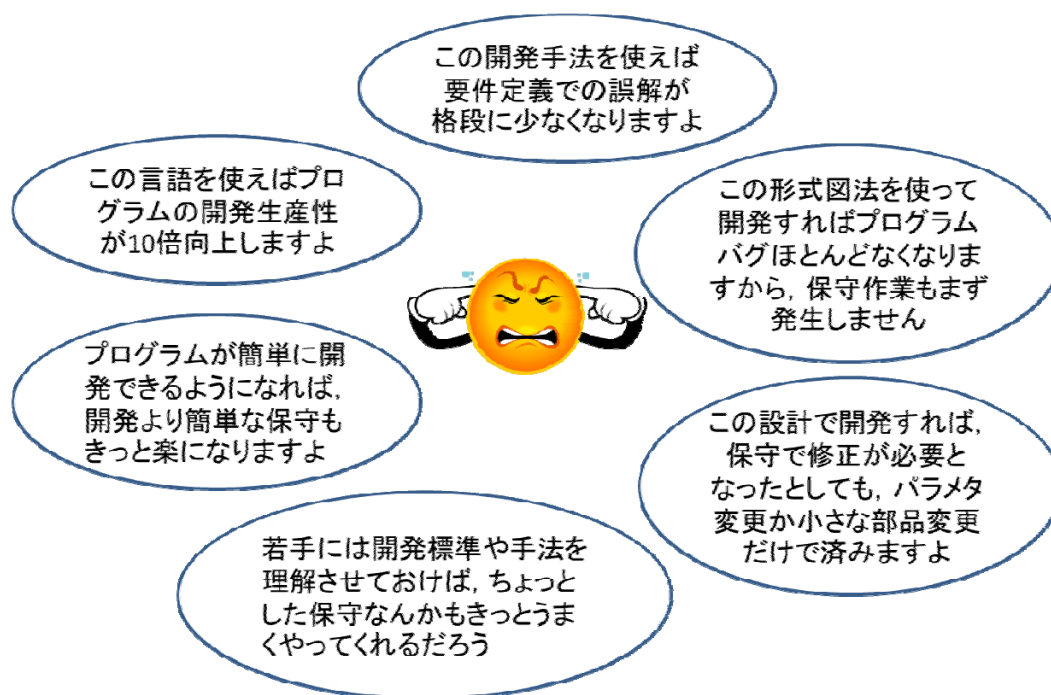
しかし、コンピュータソフトウェアが生まれて半世紀以上がたった今、その期待に近づいているといえるだろうか。論者は開発技術の進化によって保守作業を減らすことができるという夢は、実は「見果てぬ夢」ではないかと考えている。その理由は、開発され、運用に供された利用価値の高いソフトウェアは、元の姿は残し、さらに良くなることを期待されるためである。逆に利用価値のないソフトウェアはそれを求められることはない。スポーツの一流選手がファンから過去の記録を塗り替えるようなさらなる活躍を期待され、少しでも長く活躍してほしいと思われるように。

利用価値の高いソフトウェアはその運用によって多くの利益を利用者や所有者にもたらせている。その実績をベースにさらに良くすることで、より多くの利益を利用者や所有者にもたらす可能性があるとするならば、当該ソフトウェアを維持し、さらに進化させることは有効な選択肢の

一つである。また、当該ソフトウェアに潜在する欠陥や環境の変化への対応遅れで、本来得られたであろう利益がなくなることがないように手厚く見守っていくことも有益な対応となる。

このことは、あくまで既存ソフトウェアの利用価値の高い・低いが主要なパラメータであると論者は考える[5]。ソフトウェアが最近開発されたものであるとか、10年20年以上前に開発された古い技術のものかという稼働年数や採用技術年代というパラメータとは関係が少ない。開発後多くの年数が経過していても、利用価値が高いソフトウェアを古い技術で開発されたソフトウェアという理由だけでリプレースすることは経営的に有効な判断とはならない。いつまでも保守し続けている古い（レガシーな）ソフトウェアは、まだまだ利益を生み出している利用価値の高いソフトウェアと考えられるかもしれない。

ところが、利用価値の高いソフトウェアの保守を継続することが経営的にメリット大にも関わらず、ソフトウェアの関係者の多くが新規開発技術に期待する（見果てぬ夢を追い求める）あまり、本来メリットの多いはずの保守プロセスの研究がほとんどされてこなかったことは大きな社会的損失であると論者は考える[1]。



(4)開発中心の考え方の弊害

ソフトウェア保守技術の向上や保守プロセスの改善を多くのソフトウェア関係者が真剣に研究してこなかった結果、次のような事態が発生している。

- ・やむなく保守作業が必要となったとき、開発経験者でない技術者に対してスムーズに保守作業が引きつがれない。
- ・複雑なパラメータの設定変更で困っている運用者やユーザがいる。
- ・情報システム部門はクライアント OS やサーバ OS のアップグレードがされるたびに慌てて泥

繩的対策を考えている。

- ・その対策は予算化されておらず、その場しのぎで、将来を考えないもっとも安い方法で保守対応している。
- ・最新技術を使った新規開発も一部あるが、遅延なく完了せず、稼働時には不具合が頻発し、ベースラインが確定していない。そんな不安定な状態で開発担当者の一部が残り、開発時のような統制がなく保守が始まっている。
- ・稼働後のソフトウェアは、いつまでたっても修正（潜在不具合や保守対応ミスによる修正）が減らない。
- ・ソフトウェア保守体制がコスト削減策の狙い撃ちにあい、少ない人数で保守案件対応に追われ、将来の保守効率化の施策（環境整備、保守情報整備など）か打てる状況ではない。
- ・現在開発中のソフトウェアも開発コストの圧縮優先で、早く安く作ることを重要視し、保守を考えないものとなっている。

(5)保守プロセス改善に医療プロセスが参考になる

本論は、このようなギャップが存在する中、ソフトウェア保守作業についてどう考えればよいかを示したものである。

結局、ソフトウェア保守作業は今後も無くなるどころか増加傾向にあり、専門の保守者が対応すべき主要な作業であると保守作業の存在を認める。そして、保守の発生を予測し、もっとも効率のかつ的確に保守を遂行する人材を育て、保守者が周りから尊敬の念を持ってみられる存在であるようにするにはどのような保守プロセスすればよいかを考えるしかない。そのとき、ベンチマークすべきベストプラクティスは医療プロセスではないかと論者は考えている。

ところで、ソフトウェアに関することは、従来のソフトウェア工学の用語で説明すべきという考え方がある。ただし、これまでのソフトウェア工学は保守プロセスを真正面からとらえた研究は開発研究に比べてまだまだ少ないと考えられる。そのため、ソフトウェア工学の言葉よりも、医療プロセスと保守プロセスの対比により、保守プロセスの理解がより進むのではないかと論者は考え本論を書いたである。



2. 本論のアプローチ

保守者を医師，保守対象ソフトウェアを患者，保守対象ソフトウェアの所有者を患者の保護者（家族や後見人）とした場合，医療プロセスの各行為（タスク）に対応する保守プロセスの行為を洗い出す。医療行為と保守行為に「対応する」ものがあるかを見つけるアプローチを最初にとる。なお，ここでいう「対応する」という意味は厳密に同等ということではなく，同様な思考過程を採る可能性があるものとする。対応関係が非常に近い行為もあれば，多少例外があるが類似思考で行う行為もある。対応の近さはいろいろあることをご承知頂きたい。

そういった個々の行為の対応を一般的な辞書やインターネット上まの用語解説から付録に一覧としてまとめた。そこから医療行為に対する保守行為に対応可能なものが多数あることがわかる。これらを具体的プロセスに当てはめ，一連のプロセス自体も両者が類似していることを次に確認する。確認できたものに対して，保守プロセスに医療プロセスに類似した能力とスキルを必要することがイメージとしてわかる。そして，現在の保守プロセスで何が足りないのか，保守プロセスを改善するには何が必要かを具体的イメージとしてとらえていくというアプローチをとる。

3. 保守プロセスと医療プロセスの類似点

(1) 症状特定プロセス

<医療プロセス>

症状の特定が以降のプロセスを的確に進めるうえで重要な作業となる。症状は問診や検査で特定する。しかし，この行為は時として困難を伴う場合がある。たとえば，問診では患者の表現力や患者が自らの症状を伝えることが不可能な場合がある。また，検査をしても異常値が常に出るわけではなく，突発的に症状が現れるがその後落ち着く場合もある（発作）。そのため，家族からのヒアリング，精密検査や検査機器を装着して発作時の状況を再現するような検査を行うこともある。このようにして，原因となる傷病名を特定するために正確な症状を把握する必要がある。

<保守プロセス>

現象の特定が以降のプロセスを的確に進めるうえで重要な作業となる。現象はユーザヒアリングや再現テストで特定する。しかし，この行為は時として困難を伴う場合がある。たとえば，ユーザヒアリングでは，ユーザが正確な現象や発生時の状況を記憶していないことがある。また，再現テストを試みるが，現象が発生しないこともありうる。現象に遭遇したユーザ以外のユーザではどうかのヒアリング，さらに条件を変動させての再現テスト，プログラムにトラップを入れ現象発生時に動作や動作時の状況のデータを採取するなどする。このようにして，原因を特定するために正確な現象を把握する必要がある。

(2) 傷病名（原因）診断プロセス

<医療プロセス>

症状を診察し，傷病の名称（原因）を特定する。このとき，迅速かつ的確に診断するために医師は傷病による症状の現れ方のパターンを多数熟知している必要がある。症状から原因となる傷病名の候補をあげ，その中からもっとも可能性の高い傷病名を特定する。これを支援するために

医療情報システムの中に電子カルテ、傷病名などの情報をさまざまな条件で検索できる機能（エキスパートシステム）が導入されている。また、インターネット上にもさまざまな医療情報が掲示されており、信頼のおけるものは傷病名特定に活用されている。

<保守プロセス>

保守者は現象から原因を特定する。このとき、迅速かつ的確に原因調査をするために保守者はその現象パターンから原因と考えられるもの（プログラムの潜在不具合、操作ミス仕様外状況、ハードウェア・OS・ミドルウェア障害など）を多数熟知している必要がある。現象から原因の候補をあげ、その中から最も可能性の高い原因を特定する。これを支援するため、過去の問合せや対応の記録をさまざまなキーワードで検索できる機能を持つシステムが検討されている。また、インターネット上にも、ハードウェア、OS、ミドルウェアなどの操作マニュアル、FAQが掲示されており、現象から原因の切り分けに役立つ場合も少なくない。

(3)処方決定プロセス

<医療プロセス>

医師は症状の原因となる傷病名（原因）が特定されると原因を除去、損傷を受けた患部の復旧、原因による作用を抑制するなどの治療行為に移る。具体的には、投薬、手術、器具装着、食餌療法、マッサージなどの治療方法を処方箋等に記述し、薬剤師、外科医、理学療法士、管理栄養士などに依頼する（依頼せず自らが行う場合もある）。このとき、原因となる傷病に対する治療方法が複数ある場合がほとんどである。症状の重篤性、緊急対応の必要性、患者の体力、年齢、免疫力、健康保険加入有無、保険診療希望有無、金銭的な余力、家族の状況、他の傷病の存在や罹患の履歴、薬剤や食事に対するアレルギー体質の有無、薬剤の副作用、医療スタッフの負荷状況などを総合的に判断し、患者にとって最適な治療方法を選択するという高度な判断能力が医師には求められる。また、その治療方法について、医学の専門知識を持たない患者や家族の理解が得られる説明ができる能力も求められる。

<保守プロセス>

保守者は現象の原因が特定されると原因の除去、処理の足りない部分の処理補充、原因となる処理を回避する処置などの解決行為に移る。具体的には、パッチプログラムの投入、問題ないモジュールに入れ替え、現象回避モジュールの投入、運用回避提案などの解決方法を報告書などでユーザに提示する。ユーザの了解が得られた場合、自らが解決策を実行またはパートナーや運用担当者に依頼する。このとき、解決方法が複数考えられる場合がある。システム停止が不可能などの運用状況、保守契約締結の有無、ベンダー側の瑕疵責任有無、現象の重大性・緊急性・影響範囲・運用回避の可能性、保守対象システム（ソフトウェア）の技術的特殊性・修正容易性・テスト容易性・移行容易性などを総合的に判断し、ユーザにとって最適な解決方法を選択・提案するという高度な判断能力が保守者には求められる。また、その解決方法について、ITの専門知識を持たないユーザやユーザの管理者層の理解が得られる説明ができる能力が求められる。

(4)治療効果検証プロセス

<医療プロセス>

治療を施した後、期待した通りに回復しているか、投薬の副作用は出ていないか、合併症は出ていないかなど、治療の効果を確認する行為が必要となる。治療の効果が想定した通りに出ていない場合は、その原因（症状の誤認、傷病名の誤認、治療方法の選択ミス等）を再度調査し、その原因となった検討プロセスに戻り（バックトラックし）、診断や治療方法の判断に問題ないかを再確認する。この場合、医師は患者に対して正確で丁寧な情報の提示と誠実な話し合いによる合意形成（インフォームド・コンセント）を行うことも重要となる。さらに、他の医師の意見を聞くという選択肢（セカンド・オピニオン）も患者にとって必要となる場合がある。

<保守プロセス>

解決策を実施した後、期待した通りに問題現象は解消しているか、解決策の結果性能が劣化していないか、別の新たに問題を発生させていないかなど、解決策の効果を確認する行為が必要となる。解決策の効果が想定した通りになっていない場合は、その原因（現象の誤認、原因の誤認、解決策の選択ミス等）を再確認し、その原因となった検討プロセスに戻り（バックトラックし）、原因特定や解決方法選択の判断に問題ないかを再確認する。この場合、保守者はユーザに対して正確で丁寧な情報の提示と誠実な説明とユーザの疑問に答え、新たな解決策に対して合意を得ることも重要となる。さらに、別の経験豊かな保守者の意見やレビューをさらに求めることがユーザへの理解を得るために必要な場合がある。

(5)完治判断プロセス

<医療プロセス>

一回の治療で完治する場合もあるが、治療に長期間を要する場合がある。完治までの期間が必要であれば、その間患者が不安にならないようケアを継続する必要がある。治療の効果が確実に出ていること、快方に向かっていること、完治するまでは痛みやだるさが出ることがあるなど、想定された状況であることを患者が納得するよう説明する能力が医師に必要となる。

<保守プロセス>

ソフトウェアを改修したことで問題部分が改修されていても問題解決が完了したわけではない。本運用で改修したソフトウェアの処理が動作し、問題現象が解消したことを確認して完了とすべきである。年次処理で発生した問題現象は、プログラム改修が済んでも最長 1 年近く経たないと本運用で当該処理が動作しないことが考えられる。また、再現性が一定でない場合、本稼働で問題現象が発生する状態が必ずしもいつ来るか予測できない。さらに、問題現象が発生する状況となっても問題現象が解消されていれば見た目上問題現象は発生していないため、以前なら発生していた状況で今回解決したため発生しなかったのかどうか判断できない。そのため、ユーザに問題が解消したと考えても良いと判断した場合、納得性を持って問題解消をユーザに説明できる能力が保守者に必要となる。

(6) 予防プロセス

<医療プロセス>

傷病の予防が重要な行為として位置づけられている。医師や看護師は健常者に傷病に対する抵抗力、免疫力の向上、伝染病のワクチン接種、定期健康診断結果データの正常値維持を目的とした健康管理、運動療法、食餌療法などを指導する。このことにより、傷病発生の要因の除去、傷病が発生した場合の早期発見・早期治療（計画的かつ軽い対応）が可能となる。

<保守プロセス>

問題が発生してから対応する事後的な保守（是正保守、適応保守）が主と思われがちだが、予防的な保守（予防保守、完全化保守）も重要な作業として位置づけられる。保守者は保守対象のソフトウェアに対して、保守チーム内部発見の不具合情報、運用上の動作環境の将来的変化（負荷集中、システム構成の変化、不正アクセス、ハードウェア・OS・ミドルウェアの不具合、ユーザの使い方やニーズの変化、社会情勢の変化、法律の改正等）の予測情報から、現状のソフトウェアが耐えられない部分や修正に多くの工数を要する部分について、計画的な先行対応を検討する。このことにより、事象発生時の影響を最小限にすることができる。

(7) 延命治療プロセス

<医療プロセス>

医師は生命予後不良で根治が見込めない患者に対し、人工呼吸や輸血、輸液などによって延命を図ることがある。生命尊厳から延命の努力は自然ではあるが、患者本人の苦痛が継続または増大し続ける、家族の精神的苦痛や経済的苦痛の増大、制度的な医療費の増大など、さまざまな生命観、倫理観、価値観を理解し、どの時点まで延命治療行うか、人間としてどうあるべきかといった高度な判断を医師は求められる。

<保守プロセス>

保守者は保守対応が困難になっているソフトウェアに対し、暫定的な修正対応や正常に動作する部分のみ処理が走るよう制限を伏すなどをして、延命を図る場合がある。ソフトウェアは生命体ではないため、延命措置は（倫理的な要素ではなく）経営的な判断で延命が図られることが多い。たとえば、再開発やリプレースの予算が確保できない場合や連携するシステムとの関係で現状のソフトウェアを維持する必要があるなど。保守者は保守を継続する機会損失額（将来にわたる保守工数の増大、事故時の対応の遅れによる損害増大、技術者のモチベーション低下による生産性減など想定以上のコスト増）を明らかにして、再開発またはリプレースの予算との冷静な比較判断をユーザに促す必要がある。

(8) 終末医療（ターミナルケア）プロセス

<医療プロセス>

患者の傷病に回復の見込みがなく、余命が限られている場合、身体的苦痛や精神的苦痛を軽減することによって、人生の質、クオリティ・オブ・ライフ（QOL）を向上することに主眼が置かれ、医療的処置（緩和医療）に加え、精神的側面を重視した総合的な措置がとられる。医師はさ

さまざまな終末期を迎えている患者や家族に対して、どのような措置を講ずることが良いか、判断できる能力と知識を求められる。この判断を的確にできるには、多くの終末期の患者を臨床するという経験が必要となる。若手の医師はベテラン医師の指導の下に経験を積むことが重要である。

<保守プロセス>

ソフトウェアのリプレースの時期（保守期間終了）が明確になっている場合、その余命（保守終了までの期間）によって、保守の行為に一定の制限が発生する。たとえば、保守終了時期までに終わらないような保守対応、保守終了時期までに対応が終わったとしても運用上当該問題現象が発生しない保守対応、何とか運用でカバーでき運用回数が僅かな問題現象への保守対応、大幅な機能改善対応、新プラットフォームサポート対応、リファクタリング、ドキュメントの整備などは積極的に行う必要は少ないことを保守者は知っておく必要がある。実施の必要性判断について的確にできるには、多くの終末期のソフトウェア保守の経験が必要となる。経験の浅い保守者はベテラン保守者の指導の下に経験を積むことが重要である。

4. 保守プロセスの課題解決に有効な考え方が医療プロセスにあるか？

保守プロセスではさまざまな問題（未成熟な部分）を抱えている。それらの課題を解決する方法として、医療プロセスが参考になるかを検討する。それぞれの課題解決には同じことを繰り返して説明している部分があるが、上から順番ではなく個々に独立して読めるよう、あえて重複部分を残している。

(1) 開発プロセスに比べて保守プロセスは魅力が感じられない

保守プロセスを医療プロセスに当てはめるなら、開発プロセスは何になるだろうか。ヒトに当てはめると、丈夫で頭の良い子を産む、または社会で立派に働けるまで育てるというプロセスに当たるかもしれない。ヒトが居なければ医療プロセスの多くは存在の必要がないこととなる。しかし、丈夫で頭の良い子を産むことや社会で活躍できるまで育てる行為と、生まれた後死ぬまでの医療行為とどちらが上であるか下であるか比較することは意味がないだろう。なぜなら、どちらも今のヒトが暮らすうえで不可欠なプロセスだからである。

しかし、保守プロセスと開発プロセスを比較するとほとんどのIT技術者は「開発をやりたい」という状況が半世紀以上前から変わっていない。その状況は医師や教師がいない時代（石器時代等）に親や祖父母や村の長（おさ）が子供の医療や教育（躾）を試行錯誤で行い、回復の見込みがないと子供の育児放棄や子供の死を待つだけの時代と同じ成熟度といえるのではないか。その後の時代になって、医療プロセスを専門に行う医師や看護師が現れ、医療技術が進化していったように、保守プロセスを専門に行う保守者が現れ、ソフトウェアの保守技術が進化することは時代の流れと見るべきだろう。

(2) 保守プロセスが定着していない

多くのソフトウェア担当部門では、ソフトウェアの開発プロセス標準はあるが、保守プロセス標準を開発プロセス標準と同等またはそれ以上に整備しているところは少ないのではないか。そ

の理由は、ソフトウェアはまず開発ありきという固定観念があり、開発自体もうまくいっていないのに、保守プロセスの標準まで手が回らないためだろう。保守は開発部門が掛け持ちで行うということが、もっとも効率的という固定観念がそうさせている部分もある。しかし、開発と名の付いた部門に所属する技術者が保守を真正面から考えて組織的に対応することを求めるのは無理がある。ヒトを生む行為や初頭教育する人材と生まれた後の医療行為を実施する人材が別になることで、医療プロセスが確立されたように、開発部門と保守部門を分けることで、保守プロセスは保守部門内に自然と確立されると論者は考える。

(3) 保守の見積について標準ガイドラインがない

多くのソフトウェア担当部門では、開発見積標準はあるが、保守の見積標準がさまざまな保守のタイプごとに整備されているところは少ないと思われる。その理由は、保守プロセスを忠実に実施する工数を見積もる場合、修正ボリュームに対して、影響度調査、修正設計、テストの工数が多くなること（フタコブラクダ現象）が一般的である。しかし、開発プロセス中心の見積発想（ヒトコブラクダ）からではその見積は受入れがたく、整理されてこなかったと考えられる。では、保守作業はどう見積もられてきたか。一部の塩漬けとなった保守専任者が、保守案件に対し勘と経験で修正をし、テストは期間とコストの範囲内で一部だけ行えばよいとの判断を担当者任せで進めてきたからである。その方法で多くの保守案件は対応できてきたが、限られた塩漬けメンバーが対応することで、ブラックボックス化、暗黙知化が進む。専任メンバーがリタイアした場合や大きな保守上の問題が起こった際に、タイムリーな組織的な対策が打てないことになる。このような不完全な保守プロセスを前提とした見積が行われている状況は、安いからといって医療プロセスを無視するような医師（無資格医師）を前提に医療費を考えるようなものである。

(4) 保守者専門の組織がない

保守者専門の組織を持っている部門は少ない。その理由は、保守専門の組織に所属すると塩漬けになり、開発を担当できる機会が減るのではないかと、最新のITを経験する機会が減るのではないかと、キャリアパスは大丈夫か、などのメンバーの不安を増大させないことがあると考えられる。しかし、保守プロセスが確立しないこと、保守の見積や作業の標準が確立しないことは、保守組織が独立していないためであることは明らかである。医療組織が家庭、勤務先、教育機関などと独立しているように、保守専任組織を独立させることで保守の成熟度は急速向上し、保守プロセスの確立や継続したプロセス改善が実施される。

(5) 保守者専門の教育機関がない

医療プロセスも初期のころは師から弟子へ伝授されるものだったと聞く。今も医師の養成には臨床研修を始め、指導医の下でさまざまな経験を積み、医療技術を身に付けていく。そのための養成期間は医学系大学入学後10年を要することもある。その間、大学病院の医局や総合病院の研修医として実務経験を積むことが重要な教育内容となっている。保守者の養成はどうであろうか。ある開発プロジェクトを経験した若手技術者の何人かが稼働後も当該システムのソフトウェア保

守者として残り、課題（不具合、性能問題、仕様問題など）の対応を行う中で保守経験を積むことが多い（教育カリキュラムとして計画的に行われることはない）。また、保守経験の豊かな先輩が指導するようなことはまれであり、定期的に保守技術のスキル向上度合を評価されることも少ない。これでは、せっかく先輩技術者が経験から得た保守対応スキルを後輩に伝授することもなく、経年とともに保守対応の組織的な成熟度向上もない。早急に医師の養成制度を参考にして保守者としての教育プロセスを整備する必要がある。

(6) 保守者を育成する指導者がいない

保守を長年担当している技術者は、本人が自覚しているがどうにかに係らず、保守対応の生産性が保守対応経験の少ない技術者に比べて数倍以上高いことも珍しくない。このような高生産性の保守技術者は本来若手技術者を指導し、保守対応能力を組織として向上させる必要がある。しかし、保守を若手に担当させるとモチベーションが下がるという固定観念や保守は一時的作業という固定観念により、管理者は保守に必要なスキルの指導を受ける機会や指導をする機会を設けることに積極的でない。逆に、保守対応の生産性の高い技術者には、後半を育てるより少しでも多くの仕事をさせることが正しい選択と考える傾向がある。短期的にはそれがコストパフォーマンス最大の場合もあるが、リスク管理の観点からは明らかに不適切な対応であるのは間違いない。保守要件は伝染病の大流行のように予測不能な形で緊急に多発する可能性がある。そういった危機的状況で対応できるメンバーがすぐに用意できるのかを考え、保守者の育成を図る必要がある。

(7) 保守者に保守の効率化を研究する余裕が与えられない

保守者は能力やスキルの高低で生産性に数倍以上の開きがある。優秀な保守者はさらに効率的に作業を個人プレーではなく組織として実現したいと考えている。しかし、短期的な生産効率を考えている管理者（開発を主とした管理施策に偏重）は、そういった保守プロセスの改善に興味がなく、発生した保守案件を一時的なものと考え、もっとも生産性の高いベテラン集中させる。その結果、組織的な保守プロセス改善が進まない状況が続く。医療プロセスの場合、最新医療技術の学会での共有、最新医療機器のプロモーション（展示会等）、さまざまな臨床結果の収集協力依頼など将来の医療技術向上に向けた推進力が働いている。そのような状況に保守プロセスをもっていくことが求められる。

(8) 保守者の評価基準がない

保守案件を一時的な対応と考えている管理者（開発を主とした管理施策に偏重）は、保守案件で効率的に作業していることに対して、当たり前という評価に陥りがちとなる。若手に保守案件の対応をベテランと同程度アサインすれば、その差が歴然となり、評価も変わる可能性がある。しかし、保守対応を本流と考えていない管理者には保守者へ高い評価が行われる可能性は少ない。保守専門の組織ができ、保守プロセス（含む育成プロセス）が確立されることで、保守者の評価基準も確立される。ただし、医療の分野でも医師の評価基準はどうかというと、定まっていないのが現状で、医師不足が要因と考えられる。保守プロセスが確立されても、保守者不足が深刻に

なれば、評価基準が適用されない可能性がある。

(9) 保守対応の良否についての標準ガイドラインがない

医療プロセスにおいて、医療サービスの良否について評価するのは患者である。しかし、患者間の情報共有がシステムとしてなされていない以上、医療サービスの良否を統計的に評価することは困難である。医療機関の自己評価に委ねられる傾向が強い。医療過誤による医療事故についても、患者側で専門的に評価をするのは困難である。第三者機関による評価システムが必要と考えられる。保守プロセスにおいても同様である。明らかに保守対応後の動作が不良の場合は、ユーザが問題視するが、保守対応が本当に適切に行われたかどうかをユーザが専門的に確認することは容易ではない。保守対応のプロセス監査が必要である。

(10) 保守者のローテーションが円滑でない

保守では対象システムをよく知ったベテランの保守者がずっと保守を担当することが最も少コストで済む。その結果、保守者は塩漬けになり、保守者のローテーションが行われない。また、塩漬けになった保守者は組織的な情報共有をする必要がなく（しなくても怒られない、しても誰も見ない、関心がない）、保守対応履歴が暗黙知化、ブラックボックス化が進む。管理者は短期的な視点で管理すると保守者のローテーションが進まないことを自覚する必要がある。いっぽう、医療プロセスの分野でもローテーションしやすい環境かというと、そうでもない。どの病院や診療所も長年同じ患者を診続けている方が的確な治療や措置ができる。患者や医療機関から見れば、ローテーションはしてほしくない方向に行く。特に僻地の診療所では、医師の定着率を良くしたいという悩みがある。しかし、医療プロセスではカルテや使用機器に標準化が進み、医師の最低限の技能教育体制もあるため、多少の不安があっても、ローテーションへの影響は少ない。保守プロセスでは保守者育成体制を整え、必要な保守情報を形式化し、ローテーションをルーチンワーク化することが求められる。

5. 保守プロセス改善へのアプローチ

ここまでの論議で、保守プロセスの改善の検討に医療プロセスが参考になることが分かった。しかし、医療プロセスに比べて保守プロセスは成熟度が低く、ソフトウェアに係る人々の理解が進んでいる状況とはいえない。そこで、論者なりの保守プロセス改善の方法をより具体的に提案する。

(1) ソフトウェア医師の名称を広める

ソフトウェアの保守者のことを「ソフトウェア医師」または「ソフトウェア医」と呼んでみたらどうだろうか。

(2) 保守プロセスを医療プロセスの言葉に変え、分かりやすくする

専門的で分かりにくい保守プロセスを医療プロセスの用語に当てはめて試してみる。

例：

- ① プロセス実装 ⇒ ソフトウェア医院の設立準備
- ② 問題分析及び修正の分析 ⇒ ソフトウェアの診察
- ③ 修正の実施 ⇒ ソフトウェアの治療
- ④ 修正レビュー及び受入れ ⇒ 治療の結果確認
- ⑤ 移行 ⇒ 治療後ソフトウェアの復帰
- ⑥ 廃棄 ⇒ ソフトウェアの終末

(3) 新プロセス名称で事例（具代的保守行為）を当てはめる

ソフトウェア保守のタイプ別に上記(2)で示した①～⑥のアクティビティをもとに事例を示す。

例：

<緊急保守>

- ①ソフトウェアの ER 室の設置
- ②ソフトウェアの急患の症状と傷病の原因を確認
- ③原因部分の緊急除去・改修（応急・救命対応）
- ④治療効果・救命の確認
- ⑤本稼働機復帰のためリハビリ担当（運用者）へ移行・是正保守（恒久対応）への移行
- ⑥治療断念（本稼働機復帰断念）

<是正保守>

- ①ソフトウェア病院の設置
- ②ソフトウェアの患者の症状と傷病の原因を確認
- ③原因部分の計画的除去・改修
- ④治療効果の確認
- ⑤本稼働機復帰のためリハビリ担当（運用者）へ移行
- ⑥治療断念（本稼働機復帰断念）

<予防保守>

- ①ソフトウェア保健所の設置
- ②健常ソフトウェアの健康診断の実施
- ③将来症状が出る可能性のある部分の除去・改修
- ④正しく予防が終わり，副作用が出ていないことを確認
- ⑤本稼働機復帰のためリハビリ担当（運用者）へ移行
- ⑥予防断念（本稼働機復帰断念）

<適応保守>

- ①ソフトウェア病院の設置
- ②ソフトウェア動作環境の激変に耐えうるか診断する
- ③激変に耐えうるようにソフトウェアの肉体改造を行う

- ④正しく改造が終わり，副作用が出ていないことを確認
- ⑤本稼働機復帰のためリハビリ担当（運用者）へ移行
- ⑥改造断念（本稼働機復帰断念）

<完全化保守>

- ①ソフトウェア能力向上センターの設置
- ②将来の治療・改造に向けて必要な措置を見つけ出す
- ③措置を実施する
- ④措置が完了し，副作用が出ていないことを確認する
- ⑤本稼働機復帰のためリハビリ担当（運用者）へ移行
- ⑥措置断念（本稼働機復帰断念）

(4) 各行為に必要な保守者のスキル／能力を明確にする

保守プロセスで行う必要がある行為に対して一つずつ保守者に必要なスキル／能力を提示することはできないが，包括的に必要なスキル／能力を示す。

- a) 保守プロセスに必要なリソース（設備，要員，組織，教育，支援等）を確保できる能力
- b) 保守目的に合わせた最適な保守プロセスを確立できるスキル／能力
- c) 保守案件（患者の症状）の正確な把握ができるスキル／能力
- d) 症状に対し，その原因を正確かつ迅速に見つける（診断する）スキル／能力
- e) 原因の除去（治療）する応急的かつ最適な方法を判断するスキル／能力
- f) 原因の除去（治療）する恒久的かつ最適な方法を判断するスキル／能力
- g) 治療の効果を迅速かつ的確に検証するスキル／能力
- h) 最適性を考慮する場合，経営的最適性，技術的最適性，顧客視点の最適性など総合的な診断ができるスキル／能力

(5) スキル／能力を得るための教育シラバスを整備する

本年度の本作業部会で SWEBOK をベースとしたシラバスの試案が報告されるが，そのような試みが多面で進むようになることで必要なスキル／能力の広さが明確になると論者は考える。

(6) スキル／能力を測る基準（合格基準）を整備する

このような教育シラバスに対して，どこまでをマスターすれば，どのような保守プロセスをこなせるかの基準を整備する。その結果，保守プロセスを実行している組織の責任者は，同組織の人的リソースが保守プロセスを実施するうえで不足がないか確認することができる。

(7) 保守者の能力判定の民間認定試験制度を創設する

シラバスと合格基準が示されても組織の管理者が正確に保守者のスキル／能力を判定できるわけではない。特に保守プロセスの経験のない管理者の場合，客観的かつある種の合理性を持って合格基準に到達しているかを判断する試験制度は，他の技術者の試験制度を見ても妥当な制度であると論者は考える。

(8) 高能力認定技術者のトレード先の斡旋を行う

保守プロセスを高度にこなせる技術者を、保守プロセス担当の多くの組織に紹介し、適材適所を図っていく。医療プロセスでいう医局のような制度も必要だと論者は考える。

6. まとめ

ソフトウェアの保守プロセスの改善、保守者の待遇改善や保守者自身のモチベーションの向上に向けた対策に、医療プロセスが十分参考にできることがわかった。本来、ソフトウェアの開発プロセスの研究と同等な保守プロセスの工学的研究や評価がなされていれば、本論のようなレポートを出す必要がないのかもしれない。なぜなら、自律的に保守プロセスを行うために必要なスキル／能力が明らかにされ、スキル／能力を身につけるための教育やトレーニングの内容が検討され、高度な保守プロセス担当技術者（保守者）であることを正当に評価する仕組みですでにできているはずだからである。

ソフトウェア保守を必要悪と考えるのではなく、認知された一般的かつ必須なソフトウェア作業のひとつであることをすべてが認識して、多くの子供たちが「僕（わたし）、大きくなったらやっぱりソフトウェアの医者になりたい」という日がくるまで論者の仕事は終わらない。

参考文献

- [1] 増井他「～ISO14764 による～ソフトウェア保守開発」2007 ソフト・リサーチ・センター
- [2] ISO/IEC/IEEE14764 “Software Engineering-Software Life Cycle Processes-Maintenance”
2006 ; JIS X0161 2008
- [3] 増井「保守とはソフトウェアへの愛情をもったケアである」2012 SERC 年次活動報告書：作業部会「S E R Cの考える保守とは」活動報告
- [4] 「2011 ソフトウェア開発管理基準に関する調査報告書」2012：社団法人 日本情報システム・ユーザー協会
- [5] 増井「ソフトウェア保守の経済」2006～2010 SERC 年次活動報告書：作業部会「S E R Cの考える保守とは」活動報告

付録

医療・ソフトウェア保守対比表

No.	医療用語	分類	意味	対応するソフトウェア保守用語
1	ICU	基 盤・設備	自動的・継続的に血圧・呼吸・心電図などの観察ができ酸素吸入・人工呼吸などの救命・生命維持装置を完備した特別な病室。常時看護が可能で、重症患者を収容する。集中治療室。	特定ソフトウェアにトレースを入れ 24 時間 365 日監視体制。
2	ER	組 織・制度	救命救急室のこと。	緊急保守対応室。
3	医院	組 織・制度	医者が個人的に経営する(病院よりも小規模の)診療所。	ソフトウェア保守コンサルタント事務所。
4	医員	構成員	医療に従事する職員。医院、診療所、病院に勤務する医師。	ソフトウェア保守技術者。
5	医学	学 問・技術	生体の構造・機能および疾病を研究し、疾病の診断、治療、予防の方法を開発する学問。基礎医学、臨床医学、社会医学、応用医学などに分けられる。	ソフトウェア保守工学(まだ確立されていない)。
6	医学部	学 問・技術	大学で医学を教育・研究する部門。	ソフトウェア保守工学(まだ確立されていない)。
7	医業	ビジネス	医者の職業。	ソフトウェア保守サービス業(名称はないが実質的には行われている)。
8	医局	組 織・制度	大学病院を中心とした伝統的な医師の囲い込み、派遣制度。	ソフトウェア保守技術者派遣業(または紹介業)法人。
9	医局長	構成員	医局の長。医師の派遣元(主に大学病院)。	ソフトウェア保守技術者派遣業(または紹介業)法人の長。
10	医師	構成員	所定の資格を得て、病気の診察、治療を業とする人。医者。	ソフトウェア保守技術者。
11	医師会	組 織・制度	医師の組織団体、全国的なものとして社団法人日本医師会があり、別に日本歯科医師会がある。	ソフトウェア・メンテナンス研究会の将来の姿？
12	医師不足	課題	医師の数が、医療に必要とされる人数に比べて不足すること。	ソフトウェア保守技術者不足。
13	医師法	組 織・制度	医師全般の職務・資格などを規定する法律。	ソフトウェア保守技師法(対応するものなし)。
14	医師免許	組 織・制度	医師国家試験に合格した者の申請で医籍に登録されたもの。厚生労働大臣が免許を与えたときは免許証が交付される。	ソフトウェア保守技師免許(対応するものなし)。
15	医者	構成員	医師の通称。	ソフトウェア保守技術者。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
16	一次救命措置	行為	呼吸が止まり、心臓も動いていないと見られる人の救命へのチャンスを維持するため、特殊な器具や医薬品を用いずに行う救命処置であり、胸骨圧迫と人工呼吸からなる心肺蘇生法(CPR)、そして AED の使用を主な内容とする。	運用者によるシステムダウン回避。ソフトウェアを修正しないでシステム障害を解消する行為は運用者のミッション？
17	一般開業医	ビジネス	GP(General practitioner)の日本語訳。傷病の原因が分からない時、最初に診察を受ける病院。日本では位置づけが必ずしも明確でない。	システム保守総合ヘルプデスクセンター。
18	一般用医薬品	組織・制度	医師による処方箋を必要とせずに購入できる医薬品のことである。市販薬、家庭用医薬品、大衆薬などとも呼ばれる。	エンドユーザが利用できる既存ソフトウェアの修復ツール。
19	医薬品	概念	飲んだり(内服)塗ったり(外用)注射したりすることにより、人や動物の疾病の診断、治療、予防を行うための物。初期医療では、症状に合わせて都度調合していた。	修正モジュール。修正パラメータデータ。
20	医療過誤	課題	診断・治療の不適正、施設の不備等によって医療上の事故を起こすこと。誤診・誤療がその例。	ソフトウェア保守ミス。修正ミス、構成管理ミス、不法修正など。
21	医療機関	組織・制度	医療法で定められた医療提供施設。狭義においては、医師、歯科医師等が医療行為を行う施設である医院、病院、診療所をさす。	ソフトウェア保守サービス提供法人。
22	医療技術	学問・技術	医療行為を的確・迅速に行うことを可能にする技術。	ソフトウェア保守技術(存在はしているが体系化されていない)。
23	医療行為	行為	人間の健康の維持、回復、促進などを目的とした諸活動。	ソフトウェア保守作業。
24	医療従事者	構成員	病気や障害を持った人に、専門的知識と技術を行使し、その人がその人として生活できるよう手助けする職種。医師、看護師、理学療法士、作業療法士、介護福祉士 等々を指す。	ソフトウェア保守サービス法人の構成員(含む管理要員)。
25	医療情報	概念	カルテのこと。	構成管理情報(修正履歴付)、ソフトウェアタグ(研究中)。
26	医療ソーシャルワーカー	組織・制度	医療分野で社会福祉活動に従事する専門職。	ソフトウェア保守啓蒙活動家。保守すべきなのに放置している企業や個人を指導する人。すでにコンピュータウィルスに関しては進み始めている。
27	医療訴訟	行為	医療行為の適否や、患者に生じた死亡・後遺障害などの結果と不適切な医療行為との因果関係、さらにそのような結果に伴って発生した損害の有無及び額が主要な争点となった民事訴訟のこと。広義では、業務上過失致死傷罪の罪名のもと、医療行為上の過失の刑事責任が問われる刑事訴訟の場合も含む。	IT 訴訟(ソフトウェア保守ミス)。
28	医療知識	学問・技術	医療に必要な知識。	ソフトウェア保守及び対象業務の知識(含む一般常識)。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
29	医療費	ビジネス	保健医療機関において点数化された療養として現物給付されたものと、歯科医院や、保健医療機関以外の医療機関(鍼灸院・接骨院)において受けた医療行為に対して、一旦全額負担した後還付される療養費とがある。	ソフトウェア保守費(保守契約費、スポット保守費、保守コンサル費など)。
30	医療法	組織・制度	医療を提供する体制の確保と、国民の健康の保持を目的に、病院・診療所・助産所の開設・管理・整備の方法などを定める法律。	(対応するものなし)
31	医療保険	組織・制度	医療機関の受診により発生した医療費について、その一部又は全部を保険者が給付する仕組みの保険。	(対応するものなし)
32	医療録	組織・制度	カルテのこと。	構成管理情報(修正履歴付)、ソフトウェアタグ(研究中)。
33	インフォームド・コンセント	行為	医療行為(投薬・手術・検査など)や治験などの患者が、治療や臨床試験・治験の内容についてよく説明を受け十分理解した上で、対象者が自らの自由意思に基づいて医療従事者と方針において合意すること(単なる「同意」だけでなく、説明を受けた上で治療を拒否することにも含まれる)。	ソフトウェア保守対応の内容を丁寧にユーザに説明、保守サービスを受ける側が自らの意思で当該保守対応を受けるか判断すること。
34	ウィルス	原因	細菌とことなり細胞を持たないが、遺伝子を持ち、宿主細胞に宿主して増殖する微粒子。	コンピュータウィルス。
35	衛生検査技師	構成員	厚生労働大臣の免許を受けて、医師の指導・監督の下で微生物学的・血清学的・血液学的・病理学的・寄生虫学的検査を行うことを業とする者。	現象再現担当者。テスト系でのテスト担当者。
36	疫学	学問・技術	個人ではなく、集団を対象とした、疾病に対する統計による研究。	保守発生対応履歴分析。
37	エックス線検査	行為	X線を使って身体内部の透過像を写真にとる検査。	リバースエンジニアリング。
38	遠隔医療	行為	医師と患者が距離を隔てたところでインターネットなどの通信技術を用いて診療を行う行為。	オンライン保守、問合せ対応
39	遠隔診断	行為	遠隔医療行為のひとつ。遠隔から患者の診断を行う行為。	オンライン保守(原因調査)。
40	遠隔治療	行為	遠隔医療行為のひとつ。遠隔で患者の治療を行う行為。	オンライン保守(ソフトウェア修正)。
41	応急手当	行為	急病または外傷を受けた時などのさしあたりの手当。	応急パッチ。ソフトウェアの変更を伴わず、あらかじめ決められた手順でシステム障害を回避する行為は運用者のミッション。
42	応用医学	学問・技術	法医学など、医療技術を使った(对患者以外の)応用分野。	サイバー犯罪捜査、ソフトウェア保守ツールの評価。
43	おくすり手帳	組織・制度	、日本国において調剤薬局や医療機関にて調剤された薬の履歴(≒服用歴)をまとめた手帳のこと。	保守対応履歴書。
44	開業医	ビジネス	自ら診療所または病院を営んでいる医師、または歯科医師のこと。	ソフトウェア保守サービス事務所。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
		ス		
45	カウンセリング	行為	依頼者の抱える問題・悩みなどに対し、専門的な知識や技術を用いて行われる相談援助のことである。	ソフトウェア保守プロセス改善のコンサルティング。
46	家庭薬	組織・制度	⇒一般用医薬品	エンドユーザ用の障害自動自己修復ソフトウェア(今はない)。
47	カルテ	組織・制度	医療に関してその診療経過等を記録したもの。	構成管理情報(修正履歴付)、ソフトウェアタグ(研究中)。
48	癌	原因	増殖が非可逆的かつ速やかで、周囲組織への浸潤、遠隔部への転移により、病巣を拡大し、生体の消耗を招来する腫瘍。	コンピュータウィルス、ある種の性能バグ。限界値制御バグ。
49	看護過程	組織・制度	ヘルスケア、看護ケアを必要としている患者や病人に、その人に可能な限り最良で、最善のケアを提供するためにどのような計画、介入援助が望ましいかと考えて、計画、行動していく一連の思考と行動の経緯のこと。アセスメント、診断、計画、介入、評価の5段階。	ソフトウェア保守プロセス。
50	看護記録	概念	看護師が行う看護活動を記録したもの。	保守対応履歴書。
51	看護師	構成員	医療、保健、福祉などの場において以下の事柄を行う医療従事者の呼称である。 ・医師等が患者を診療する際の補助 ・病気や障害を持つ人々の日常生活における援助 ・疾病の予防や健康の維持増進を目的とした教育	保守技術者(助手)。
52	看護師免許	組織・制度	看護行為を官から許可する資格。	情報処理試験(システム運用、アドミニストレータ)。ソフトウェア保守技術者試験(今はない)。
53	患者	対象	なんらかの健康上の問題のため、医師ないし歯科医師や専門の医療関係者の診断や治療、助言を受け、(広義な意での)医療サービスの対価を払う立場にある人。	問題または修正依頼のある保守対象ソフトウェア。
54	基礎医学	学問・技術	医学の研究・教育・実践上の専門分野のうち、無直接患者の診察に携わらないものの総称。	ソフトウェア保守工学(基礎理論)(まだ確立されていない)。ソフトウェア保守研究所で研究。
55	急患	対象	救急患者のこと。	緊急に問題の解消を求められている保守対象ソフトウェア。
56	救急医療	学問・技術	人間を突然に襲う外傷や感染症などの疾病、すなわち「急性病態」を扱う医療である。	緊急に問題の解消を求められている保守対象ソフトウェアに対する保守対応工学または技術。
57	救急患者	対象	急病または重傷を負い、直ちに救命または治療が必要な患者	緊急に問題の解消を求められている保守対象ソフトウェア。
58	急性病態	概念	人間を突然に襲う外傷や感染症などの疾病。	緊急重大障害。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
59	救命救急室	組織・制度	24 時間・365 日全ての救急患者(救急車来院および独歩来院)を受け入れ、一義的に ER ドクター(ER 専門医)によって全ての科の診断および初期治療(Advanced Triage)を行い、必要があれば各専門科にコンサルトするというシステム。	緊急ソフトウェア保守対応室。
60	救命救急センター	組織・制度	急性心筋梗塞、脳卒中、頭部外傷など、二次救急で対応できない複数診療科領域の重篤な患者に対し高度な医療技術を提供する三次救急医療機関。	緊急ソフトウェア保守対応センター。
61	勤務医	構成員	病院や診療所などの医療施設における被雇用者として診療に従事している医師のこと。	ソフトウェア保守技術者(会社員)。
62	怪我	原因	外的要因による組織または臓器の損傷の総称。	何らかの要因で一部破壊された保守対象ソフトウェア
63	外科	組織・制度	身体の総称や内臓諸器官の疾病を手術的方法で治療する。	パッチ。
64	外科医	構成員	手術によって創傷および疾患の治療を目指す臨床医学を担当する医師。	何らかの要因で一部破壊された保守対象ソフトウェアの修復または機能不足・不良を補う修正を行うソフトウェア保守技術者。
65	劇薬	概念	薬事法で定められ、致死量が経口投与で体重 1kg あたり 300mg 以下、皮下注射で体重 1kg あたり 200mg 以下の医薬品。	クリティカルパッチ。
66	血液検査	行為	採血法によって得られた血液を利用して病状などを調べる臨床検査の一つ。検査は主に臨床検査技師が行う。	ログ検査。
	原因療法	行為	症状や疾患の真の原因となっているものを直したり取り除いたりする治療法で、対症療法と対置される概念である。	根本対応。恒久対応。完全化保守。
68	健康診断	行為	病気の予防・早期発見などのために医師が行う診断。	定期稼働状況診断(実行ログ、エラーログ、リソース消費状況、ヘルプデスク問合せ記録、ユーザヒアリング等)。
69	健康増進法	組織・制度	国民の健康維持と現代病予防を目的として制定された法律。健康維持を国民の義務としており、自治体や医療機関などに協力義務を課しているなどの特徴がある。	稼働中ソフトウェア品質強化促進法(今はない)。 稼働中ソフトウェアの所有者は、発見された潜在不具合の是正、法律改正や動作プラットフォーム変化など稼働環境の変化への対応を迅速に行うことを義務付ける法律。
70	健康相談	行為	医師・保健師・栄養士などに健康の維持・増進などについて相談すること。	稼働中ソフトウェアの健全性診断。
71	健康体	概念	疾病がなく、栄養状態も良く、かつ身体臓器の機能も正常な身体。	品質に問題なく稼働している既存ソフトウェア。
72	健康保険	組織・制度	常時 5 人以上の従業員を使用する事業所または法人の事業所に適用される被用者医療保険。	(対応するものなし)
73	健康保険法	組織・制度	健康保険を規定した法律。保険の運営は政府が召喚するものと健康保険組合の管掌するものがある。保険料は被保険者と事業主が折	(対応するものなし)

No.	医療用語	分類	意味	対応するソフトウェア保守用語
			半で負担。	
74	言語聴覚士	組織・制度	音声機能、言語機能又は聴覚に障害のある者についてその機能の維持向上を図るため、言語訓練その他の訓練、これに必要な検査及び助言、指導その他の援助を行うことを業とする者。リハビリテーション専門業のひとつ。	パラメータチューニング担当者。
75	検査	行為	身体検査のこと。身体の発育状態及び異常の有無を調べること。	再現テスト、原因切り分けテスト。
76	検査結果	概念	身体検査項目ごとの結果。	再現テスト報告書、テスト成績書。
77	検査項目	概念	身体検査の項目	テスト仕様書。
78	研修医	構成員	臨床研修期間中の「医師または歯科医師」の呼び名。	ソフトウェア保守技術者(研修期間中)。
79	健診	行為	健康診断のこと。	原因切り分け。
80	公衆衛生	概念	組織された地域社会の努力を通して、疾病を予防し、生命を延長し、身体的、精神的機能の増進をはかる科学・技術。具体的には、疫学、生物統計学、医療制度、環境・社会・行動衛生、職業衛生、食品衛生など。	組織として、稼働中ソフトウェアの問題を早期に発見し、組織一体となって対策を立て、解決を図っていく仕組み(ソフトウェア保守者(専門家)指導のもとに)。
81	高度救命救急センター	組織・制度	救命救急センターのうち特に高度な診療機能を提供するものとして厚生労働大臣が定めるものであり、広範囲熱傷や指肢切断、急性中毒等の特殊疾病患者に対する救急医療が提供される。	高度緊急ソフトウェア保守対応センター。
82	効能効果	概念	医薬品のききめと効いた結果。	パッチ適用やパラメータチューニングの品質評価と稼働評価。
83	公立病院	組織・制度	地方公共団体が経営する医療機関。自治体病院とも言う。	公立ソフトウェア保守センター(今はない)。
84	国立病院	組織・制度	日本の厚生労働省が直接経営している施設等機関だったが、2004 年から国立病院機構に移管された。。	国立ソフトウェア保守センター(今はない)。
85	国立病院機構	組織・制度	日本の厚生労働省所管の特定独立行政法人で、医療機関等を運営する統括組織。	国立ソフトウェア保守センター(今はない)。
86	誤診	課題	医師が診断を誤ること。また、その診断内容。	原因切り分けミス。
87	混合診療	行為	日本の医療における保険診療に保険外診療(自由診療)を併用すること。	公的ソフトウェア保守保険(今はない)とスポット保守の混合対応。
88	細菌	原因	医学的には真正細菌のこと。いわゆる細菌・バクテリアと呼ばれ、大腸菌、枯草菌、シアノバクテリアなどを含む生物群である。	バグ、コンピュータウィルス、エージェント、デーモン。
89	採血	行為	病気の診断や輸血などのために血液を静脈又は皮膚切創から採取すること。	ログ、トレース、ダンプ採取。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
90	採血法	学問・技術	血液検査・血液培養検査などの臨床検査を行う上で重要な医学的手法の一つで、生体の血液を採取する方法である。一般には、静脈から採血する静脈採血と、動脈から採血する動脈採血に分けられる。	ダンプ、ログの採取法。
91	採尿	行為	検査のために尿を採ること。	削除データ、ガーベージデータ採取。
92	作業療法士	構成員	医療従事者（メディカルスタッフ）の一員であり、理学療法士（PT）、言語聴覚士（ST）、視能訓練士（ORT）と共に、リハビリテーション職と称されるうちの一つ。身体障害者と精神障害者の応用動作能力と社会的適応能力を回復させることが目的。手工芸（折り紙、木工、陶芸、編み物等）や芸術（音楽、絵画、塗り絵、書道、俳句等）、遊び（トランプ、将棋、オセロ、パズル等）やスポーツ（散歩、体操、ゲートボール、ダンス等）等の「創作活動やレクリエーション」、日常動作（食事、料理、掃除、読書等）である「生活活動」等の「行為（作業）」を通し、次の段階である「社会復帰する為の訓練」をさせて、日常生活をスムーズに送るための複合的な動作が可能になるよう、リハビリテーションを行う。	ソフトウェア（プログラム、パラメータ、ドキュメント）改善作業。
93	サプリメント	行為	栄養補助食品。一般にはタブレットやカプセルに特定の成分を濃縮して詰めたものをいう。正確な規定はないが、「バランスのとれた食生活が困難な場合などに、ビタミン、ミネラルなど不足しがちな栄養成分を補給したり、健康を維持するために用いられる食品」とされる。	キャパシティ増強パラメータ。
94	産業医	構成員	労働者の健康管理に当たる医師。労働安全衛生法上、一定規模以上の事務所ではこれを選任する義務がある。	ユーザ専任ソフトウェア保守技術者。
95	歯科医師免許	組織・制度	歯科医師法第2条、第6条の規定により、歯科医師になるための国家試験に合格し、その後歯科医籍に登録し、厚生労働大臣から受けることができる免許。	（対応するものなし）
96	自己診断	行為	診断リストに基づいて、自分の健康に問題がないかチェックし、医師等が作成した判断シートで結果を診断する。	例外処理、動作不安定メッセージ、管理者コールメッセージ。
97	疾病	原因	身体の諸機能の障害。健康でない異常状態。	バグ、コンピュータウィルス。
98	指導医	構成員	高度な知識や技量、経験を持ち、認定医や専門医などを指導する立場にある医師・歯科医師として学会が認定した医師・歯科医師。	ソフトウェア保守工学・プロセス・実作業の指導者。
99	死亡診断書	組織・制度	死亡事由などについての検案について記した書類で診断書の一つで、診断した医師もしくは歯科医師のみが死亡診断書を発行できる。	ソフトウェア廃棄確認書。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
100	社会医学	学 問・ 技術	生物としての人間だけでなく、社会的存在としての人間(集団)を重視して研究、診療を行う医学。	社会保守学(今は存在しない) ソフトウェアを単なるコンピュータ上での存在と見なさず、社会的存在として重視する研究・保守する学問。不具合があっても、社会的観点から修正しない選択肢もありうるとする。
101	社会福祉士	組 織・ 制度	社会福祉士は社会福祉士の名称を用いて、専門的知識及び技術をもって、身体上若しくは精神上の障害があること又は環境上の理由により日常生活を営むのに支障がある者の福祉に関する相談に応じ、助言、指導、福祉サービスを提供する者又は医師その他の保健医療サービスを提供する者その他の関係者との連絡及び調整その他の援助を行うことを業とする者。名称独占資格。ソーシャルワーカーとも。	(対応するものなし)
102	社会保障制度	組 織・ 制度	個人的リスクである、病気・けが・出産・障害・死亡・老化・失業などの生活上の問題について貧困を予防し、貧困者を救い、生活を安定させるために国家または社会が所得移転によって所得を保障し、医療や介護などの社会サービスを給付する制度。	(対応するものなし)
103	自由診療	ビ ジ ネ ス	健康保険の適用を受けず、医療機関で掛かった費用の全額(健康保険負担額無し)を支払って受ける診療。	スポット保守対応。
104	手術	行為	用手的に創傷あるいは疾患を制御する治療法で、切除、形成、移植、検査等のために生体に侵襲を加える医療行為。	大規模なソフトウェア修正、リファクタリング。
105	症状	概念	病気の状態。	現象。
106	常備薬	行為	常に手元に置いておく薬。	保守用ツール。
107	傷病	原因	負傷と疾病。	バグ、コンピュータウィルス。
108	触診	行為	手で触って異常がないか調べる。手触り、温度、硬さ、弾力、腫瘍の有無、圧痛の有無など、様々な所見がとられる。	ログ、トレース、ダンプ採取、デバックモード実行、コードレビュー。
109	処方箋	組 織・ 制度	診療所や病院などの医療機関を受診した結果、医師、歯科医師、獣医師が作成(処方)し、投与が必要な医薬品とその服用量、投与方法などを記載した薬剤師に対する文書。	修正仕様書。
110	私立病院	組 織・ 制度	私人がその費用で設立し、維持する病院。	ソフトウェア保守専門企業。
111	人工呼吸	行為	自発呼吸が不十分な人に対し、人工的に呼吸を補助すること。	セーフモード実行。
112	診察	行為	医師が患者のからだを調べて、病状・病因などを探ること。	ログ、トレース、ダンプ採取、デバックモード実行の調査。
113	診断	行為	医師が患者を診察して病症を判断すること。	不具合原因特定。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
114	診療科	組織・制度	病院や診療所などにおける医療における診療の専門分野区分のこと。旧来は内科、外科、耳鼻咽喉科、眼科、泌尿器科などの分類であったが、最近では、総合診療センター、救命救急センター（ER）、消化器病センター、呼吸器病センターなどの名称や分類に変ってきている。	不具合原因の調査と改修を行うソフトウェアの分野（業務系、業種系、組込系、科学技術系など）。総合ソフトウェア保守センターなど。
115	診療所	組織・制度	医師又は歯科医師が、公衆又は特定多数人のため医業又は歯科医業を行う場所であって、患者を入院させるための施設を有しないもの又は 19 人以下の患者を入院させるための施設を有するものをいう。	不具合原因の調査と改修を行うソフトウェアの組織。
116	診療情報	概念	カルテのこと。	保守対応履歴情報。
117	診療放射線技師	構成員	病院・診療所などの医療機関において放射線を用いた検査・治療を業務とする、国家資格を有する医療職。	原因の特定と改修のためにログやトレースを採取し、解析する技術者。
118	診療報酬	組織・制度	保険診療の際に医療行為等の対価として計算される報酬を指す。	保守実施の対価。但し、請求左記はエンドユーザではなく、エンドユーザが保守契約をしているベンダー。
119	心理療法	学問・技術	物理的・化学的手段に拠らず、教示・対話・訓練を通して認知・情緒・行動などに変容をもたらすことで、精神疾患や心身症の治療、精神心理的問題・不適応行動などの解決に寄与し、人々の精神的健康の回復・保持・増進を図ろうとする理論と技法の体系のこと。	（対応するもの無し）
120	ストレス	原因	生物学的には何らかの刺激によって生体に生じた歪みの状態を意味する。	ヘビーデューティー、トラフィックの大きなバラツキ、ランダム操作などの通常運用でない入力データ投与。
121	生物統計学	学問・技術	統計学の生物学に対する応用領域で、様々な生物学領域を含む。特に医学と農学への応用が重要である。	既存ソフトウェアの継承 (inheritance) 記録と進化過程の推測。
122	精密検査	行為	細かい点までくわしく調べること。特に、健康診断で不審な点があった場合に、具体的な病状や原因などをつきとめるためにする検査。精検。	ダンプ、トレース、更新履歴などを使って行う不具合原因の詳細調査。
123	切除	行為	手術行為の一種で、身体の腫瘍、瘤、壊死部位、変形部分などを切り取ること。	不具合処理の切り離し。スキップ。
124	先端医療	学問・技術	先端・先進技術を使った医療分野。	先端・先進技術を使った保守分野。
125	専門医	構成員	認定医よりさらに高度な知識や技量、経験を持つ医師・歯科医師として学会が認定した医師・歯科医師。	高度専門保守者。
126	臓器移植	行為	疾病や外傷によって臓器が機能しなくなった場合、事故または他人の臓器を移植すること。	不具合モジュールを他システム、他機能、新規作成した正常モジュールに入換え。
127	蘇生術	行為	一度死亡した、あるいはそれに類する状態になった人間が再び生命を取り戻す技術。	再起動術。
128	退院	行為	入院していた患者が病状が回復するなどして病院から出ること。	モジュールチェックイン。本稼働機適用。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
129	大学病院	組織・制度	基本的に医学と歯学における分野において、以下の「教育」「臨床」「研究」の3つの機能を持ち、組み合わせられて実践されている。 教育 - 基礎教育, 臨床教育, 研究教育 臨床 - 実際の医療, 臨床教育, 臨床研究 研究 - 基礎研究, 臨床研究, 臨床試験 上記の「臨床教育, 臨床研究, 臨床試験」の部分を担うために大学におかれる附属施設の1つが「大学病院」である。	大学に設置したソフトウェア保守研究センター。研究のため、外来や緊急の保守案件の対応も行う。(現状は存在しない)
130	対症療法	行為	表面的な症状の消失あるいは緩和を主目的とする治療法。	原因不明だが、現象を発生させないようにする保守対応。暫定対応の一種。
131	代替医療	概念	通常医療を補完する医療。漢方、鍼灸、マッサージ、心理療法などのこと。	遠隔操作のコンサル、運用回避提案、不具合回避のためのリソース拡張提案。
132	打診	行為	手や器具でたたいて調べる。胸部を指でたたいて反響音を確認めたり、関節の近くをハンマーでたたいて反射を確認めたりする診察行為。	高負荷実行実験、意地悪テスト、限界テスト。
133	調剤	行為	医師・歯科医師・獣医師から発行された処方箋に基づき、薬剤師が医薬品を交付すること。	修正実施。
134	治療	行為	病気やけがをなおすこと。病気を治癒させたり、症状を軽快にさせるための行為。	保守対応(修正)。
135	通院	行為	病院、医院、診療所などに患者が通って診察・治療を受けること。	持込ソフトウェアの保守依頼。
136	通常医療	概念	医療制度して認定された医療行為。その他の医療行為を代替医療と呼ぶ。	ソフトウェアの修正を伴う保守対応。
137	定期健診	行為	定期的に実施する健康診断。	定期稼働状況診断(実行ログ、エラーログ、リソース消費状況、ヘルプデスク問合せ記録、ユーザヒアリング等)。
138	電子カルテ	組織・制度	従来医師・歯科医師が診療の経過を記入していた、紙のカルテを電子的なシステムに置き換え、電子情報として一括してカルテを編集・管理し、データベースに記録する仕組み。	保守履歴記録・検索システム。ソフトウェアタグ。
139	点滴	行為	正式には点滴静脈注射と呼び、ボトルやバッグに入れて吊した薬剤を、静脈内に留置した注射針から少量ずつ(一滴ずつ)投与する方法。	(対応するもの無し)
140	伝統医学	概念	現代の医学が発達する以前から存在する世界各地の文化圏伝統の医学体系の総称。	旧技術や経験に基づくソフトウェア保守対応。
141	統合医療	概念	通常医療と補完代替医療を併せた概念。	是正保守＋完全化保守。
142	投薬	行為	傷病に適した薬剤を与えること。薬剤投与。	修正プログラム適用。
143	登録医	構成員	学会に登録している医師・歯科医師でまだ認定を受けていない医師・歯科医師。	登録保守者。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
144	ドクターヘリ	組織・制度	救急医療用の医療機器等を装備したヘリコプターであって、救急医療の専門医及び看護師が同乗し救急現場等に向かい、現場等から医療機関に搬送するまでの間、患者に救命医療を行うことができる専用ヘリコプター。	緊急時、緊急対応保守者の移動ヘリ。保守対象システムとネットワークで接続され、現象の確認、応急対応ができる設備を持つ。
145	毒薬	概念	薬事法で定められ、急性毒性における致死量が、経口投与で体重 1kg あたり 50mg 以下の医薬品。	クリティカルパッチ。
146	トリアージ	行為	人材・資源の制約の著しい災害医療において、最善の救命効果を得るために、多数の傷病者を重症度と緊急性によって分別し、治療の優先度を決定すること。識別救急。	保守案件の対応優先度付け。
147	トリアージタグ	組織・制度	トリアージを行う際、治療優先度を識別するため患者につけるタグ。	保守案件の対応優先度データ。
148	内科	組織・制度	主に身体の臓器(内臓)を対象とし、一般に手術によらない方法での診療とその研究を行う医学の一分野。	プラットフォームソフトウェアや共通関数ソフトウェアの保守を専門とする分野。
149	内科医	構成員	主に内科を専門とする医師。	プラットフォームソフトウェアや共通関数ソフトウェアの保守を専門とする保守者。
150	二次救命措置	行為	病院など設備の整った環境で、広範な患者にたいして有資格者により行われる救命処置。	(対応するもの無し)
151	入院	行為	患者が治療・検査を受けるために一定期間病院に入ること。	保守対応中のソフトウェア。
152	尿検査	行為	尿についての多くの検査項目を含み健康診断の最も一般的な方法の一つ。	ダンプ、ログ解析。
153	人間ドック	組織・制度	短期間入院して、全身の精密検査を行い、疾病の早期発見、健康指導などを行うこと。また、その施設。	稼働中ソフトウェアの定期的な詳細健全性診断。
154	認定医	構成員	高度な知識や技量、経験を持つ医師・歯科医師として学会が認定した医師・歯科医師。	認定保守者。
155	PTSD	症状	心的外傷後ストレス障害。危うく死ぬまたは重症を負うような出来事の後起こる、心に加えられた衝撃的な傷が元となる、様々なストレス障害を引き起こす疾患。	(対応するもの無し)
156	皮膚移植	行為	ドナー部位、またはドナーの皮膚を採皮し創傷などへ貼付縫合する外科手術。	画面・帳票定義ファイルの復旧。
157	肥満	原因	正常な状態に比べて体重が多い状況、あるいは体脂肪が過剰に蓄積した状況。	拡張性、保守性を設計として過剰に取込み過ぎ、効率性が低下している状態。
158	病院	組織・制度	病人を診察・治療する施設。医療法の規定では 20 人以上の入院施設を備えていることが条件。	総合ソフトウェア保守センター。
159	病因	原因	病気の原因。	ソフトウェア障害のまたは仕様未達の原因。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
160	病気	原因	生物の全身または一部分に生理状態の異常を来し、正常の機能を営めず、また諸種の苦痛を訴える現象。	ソフトウェア不具合の発生。
161	病室	組織・制度	病人のいる部屋。病院の患者を収容する施設、	チェックアウト状態のモジュール。
162	美容整形	行為	容貌・用紙を美しくするためにほどこす形成外科の一分野。	使用性向上。
163	病巣	原因	病に侵されている箇所。病原のある箇所。	不良箇所。
164	プラシーボ	学問・技術	偽薬。偽物の薬の事である。成分としては、少量ではヒトに対してほとんど薬理的影響のないブドウ糖や乳糖が使われる事が多い。	再現性が無く、原因特定が難しい状況ではあるが、ユーザは原因特定と改善を実施するよう強く要請されるとき、架空の原因を説明し、実際には何もしないパッチを適用し、客先要求を満足させる。
165	ペインクリニック	概念	主として疼痛を主訴とする疾患の診療部門であり、神経ブロックによる治療を中心に行う医療機関である。基本的には麻酔科の医師が行う。	現象緩和策。
166	防衛医療	概念	主に医療過誤の賠償責任や刑事責任追及等にさらされる危険を減ずるための、医療者側の対応として行う医療行為、あるいはリスクの高い患者の診療の忌避を意味する。	根本的な修正ではなく、安全サイドに逃げる修正⇒コードクローンなどの原因。また、安易な運用回避提案。
167	縫合	行為	外科手術・外傷などにより切断または離断された組織の機能をこと。回復させ、その治癒を促進するため、幹部を縫い合わせる事。	ラッピングモジュール。
168	補完医療	概念	代替医療のこと。	遠隔操作のコンサル、運用回避提案、不具合回避のためのリソース拡張提案。
169	補完代替医療	概念	補完医療と代替医療を併せた概念。	遠隔操作のコンサル、運用回避提案、不具合回避のためのリソース拡張提案。
170	保健所	組織・制度	地域住民の健康や衛生を支える公的機関の一つであり、地域保健法に基づき都道府県、政令指定都市、中核市その他指定された市又は特別区が設置する施設。	予防保守センター。
171	保険診療	組織・制度	健康保険に適合した診療。健康保険加入者には補助があり、個人の負担が少なくなる。	保守契約者への保守の SLA に沿った保守対応。
172	保険適用	組織・制度	現行保険の適用を受け、少ない個人負担金で医療が受けられること。	保守契約内作業。
173	発作	概念	病気の症状が急激に発して、比較的短い時間にさること。	現象が急激に発して、比較的短い時間にさること。
174	麻酔	行為	薬物などによって人為的に疼痛をはじめとする感覚をなくすこと。これにより、手術を受けることができ、また、耐え難い苦痛を取り除くことができる。	(対応するもの無し)
175	民間療法	概念	古来、民間で発見されて伝承されてきた方法によって行う病気の治療法。木や草を用いるもの。温灸・指圧・食餌療法などさまざま。	感と経験による保守対応。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
176	問診	行為	病歴・症状などを質問して診断の助けとすること。	現象ヒアリング。
177	薬剤師	構成員	「調剤, 医薬品の供給その他薬事衛生をつかさどることによって, 公衆衛生の向上及び増進に寄与し, もって国民の健康な生活を確保する任務者。」	ソフトウェア修正作業者。
178	薬事法	組織・制度	日本における医薬品, 医薬部外品, 化粧品及び医療機器に関する運用などを定めた法律。	(対応するもの無し)
179	薬理作用	概念	薬品によっておこる生理的变化。	ソフトウェア修正による処理内容の変化。
180	薬理学	学問・技術	生体に一定の科学的物質を与えた時に起こる生体现象の変化を研究する学問。	修正ソフトウェアが既存機能に与える影響を分析する学問。
181	薬局	組織・制度	薬剤師が販売又は授与の目的で調剤の業務を行う場所のこと。	修正モジュールを提供する場所。
182	輸血	行為	血液型の合う健康者の血液を患者の血管内に注入すること。外傷または手術による失血, 胃腸内出血, 貧血, 衰弱, 種々の伝染性疾患などの場合に行う。	バッファエリア拡大, 桁数拡張, 等。
183	輸血反応	症状	輸血による生体の反応, 特に副作用。溶血, 発熱, アレルギー反応, 感染など。	バッファエリア拡大, 桁数拡張, 等によるレスポンス低下(例)。
184	予防医学	学問・技術	疾病の発生・経過・分布・消長とそれに影響をおよぼす原因を研究し, 疾病の予防を行うことや, 病気になりにくい心身の健康増進を図るための学問。狭義には, 「病気になってしまってからそれを治すことより, 病気になりにくい心身を作る。病気を予防し, 健康を維持する」という考え方に基づいている医学。	予防保守。
185	理学療法	行為	マッサージ・温熱・電気などを用いる物理療法と, 筋力増強。機能訓練・歩行訓練などの運動療法とを組み合わせ, 運動障害の回復・改善をはかる治療法。	(対応するもの無し)
186	理学療法士	構成員	厚生労働大臣の免許を受けて, 医師の指示のもとに, 理学療法を行うことを業とするもの。	(対応するもの無し)
187	リハビリテーション	行為	身体的, 精神的, 社会的に最も適した生活水準の達成を可能とすることによって, 各人が自らの人生を変革していくための手段を提供していくを旨とし, 且つ時間を限定した過程である。	移行前運用テスト。
188	臨床	行為	医学・歯学・看護学等の医療分野において, また最近では心理学・教育学・社会学・法学等の学問領域においても, 医療・教育・カウンセリングその他の介入を行う「現場」, あるいは「現場を重視する立場」を指す。	ソフトウェア保守現場。
189	臨床医学	学問・	基礎医学に対して, 患者を実地に診察・治療する医学。	ソフトウェア保守の実作業学。

No.	医療用語	分類	意味	対応するソフトウェア保守用語
		技術		
190	臨床教育	行為	医療現場を経験させつつ教育を行うこと。	OJT。
191	臨床業務	行為	医療・教育・カウンセリングその他の介入を行う「現場」の業務。	保守作業。
192	臨床検査	行為	生物学的・血清学的・血液学的・寄生虫学的・生化学的検査および政令で定められた生理学的検査。	現象再現テスト, 現象解消確認テスト。
193	臨床検査技師	構成員	国家試験により免許を受けて、医師の指導・監督の下に生物学的・血清学的・血液学的・寄生虫学的・生化学的検査および政令で定められた生理学的検査を行うことを業とする者。	テスト担当者。
194	臨床研修	組織・制度	医師免許を取得した医師が 2 年間に上級医師の指導のもとに受ける実地研修。この期間、一定の年収が保障されるが、副業は禁止。現在の研修制度のポイントは、次の通り。 1. 医師としての人格を涵養 2. プライマリ・ケアへの理解を深め患者を全人的に診ることができる基本的な診療能力を修得 3. アルバイトせずに研修に専念できる環境を整備	ソフトウェア保守者教育(OJT)。
195	臨床実験	学問・技術	医薬品の新規開発で、最初は限られた条件の人(薬を設計するとき想定した症状に理想的によく当てはまる人)を対象に試験的に導入してみて、効能・服作用を調べる行為。	修正評価版の適用。
196	臨床心理学	学問・技術	人間の心理的障害・病理の問題を、心理学的な原理や知識を総合して解決することを図り、そのための理論および技術を研究する心理学の一部門。	ソフトウェア新規開発時の運用テスト。
197	臨床心理士	構成員	患者のさまざまな心理的問題を検査・診断して心理療法を行う臨床心理学の専門家。	(対応するもの無し)
198	臨床薬理学	学問・技術	実践的な薬の使用方法を主に扱う医療分野。	(対応するもの無し)
199	羸瘦	原因	肥満の対語。病的に痩せこけた状態。	拡張性が全くない状態のソフトウェア。
200	レジリエンス	能力	ストレス抵抗性。	障害許容性(fault tolerance)。
201	レセプト	組織・制度	医療機関が健康保険組合に正規宇する診療報酬証明書。	保守契約対応報告書。

(文責：増井)

3.2. 入門ソフトウェア障害報告書ーソフトウェア障害報告書はこう書けー（高橋芳広）

はじめに

障害調査の忙しい合間をぬって書いたソフトウェア障害報告書。せっかく書いたその障害報告書が、上司から書き直しを命じられたり、お客様に破りすてられたりしたことはありませんか？

※

ソフトウェアに障害が発生すると原因究明・対策の作業が急務となりますが、その合間をぬって障害報告の準備をすることも忘れてはなりません。障害が例えお客様側の問題であることが判明したとしても、報告が不要になることは稀なことです。

そして、報告の成否が分かれる重要な要素に障害報告書があります。お客様の信頼を継続し今後の作業がスムーズになるか、または怒りを買いやらなければならない仕事量増大するかは、障害報告書の出来如何によるとも言えるのです。

では、良いソフトウェア障害報告書はどういったものなのでしょう。本書では、どうやったら良い報告書になるのか、書く上での注意点を説明します。

※筆者の知人は報告書をよく破るお客様には、高級な厚紙（コーティングされたもの）に印刷したり、無闇に枚数を稼いで厚くしたりして破れない様にしたそうです。確かにこれでも、破られないという目的は達せられましたが、それはそれで...

●対象とする読者

本書は、ソフトウェア障害報告書が必要となる背景から書き方について、初歩から丁寧に解説しています。

- ・ソフトウェア開発・保守に関わる新人の方
- ・実際に障害報告書を書くことになった中堅の方
- ・障害報告書の書き方を指導する立場の方

に読んでいただければ幸いです。

●想定するシステム

本書では緊急性の高い大規模システムでの障害報告プロセスを想定していますので、中小システムや緊急性の低い障害では、全てのアクティビティは必要ない場合があります。しかしながら、全く異なったプロセスではなく、共通する部分も多いと考えますので、担当するシステムに該当する部分を参考にして頂けると幸いです。

1. ソフトウェア障害報告書とは

1. 1 ソフトウェア障害報告書の目的

ビジネス文書には必ず明確な目的があります。その目的を達成することのできるものが良い文書になります。ではいったいソフトウェア障害報告書の目的なんなのでしょうか？次にソフトウェア障害報告書の目的を示します。

(1) 情報の共有・意識合わせ

システムに障害が発生したときには、現場は混乱しています。そのため、当然のように情報が正確に漏れなく伝わる訳ではありません。このため、情報の質と量を合わせる事が重要になります。

障害調査をする組織（人）が得ている情報を提示し、正確かつ漏れがないことを確認して頂くと共に、逆にお客様側で掴んでいない情報を共有して頂く必要があります。

お客様はどのくらい困っているのか、すぐ解決しそうなのか、調査にどれだけ時間がかかるのかの情報を交換し、情報収集よりも再稼働を優先するのか、原因究明を優先するのか、エンドユーザーへのアナウンスをどうするのか等方針を合わせ、今後の対応の意識を合わせることが重要となります。また、直ぐに解決するだろうなどと、お客様に過剰な期待を抱かせないことにも注意が必要です。

(2) 協力して問題を解決する体制を作る

障害の原因を究明し、対策を滞りなく行うためには、運用者と保守者、他社ベンダー、関係者全員の協力が不可欠です。システム障害で混乱していると、「保守者は親身になって調査を行ってくれているのだろうか？」とか、「未だ得られてない情報や隠されている情報があるのではないか？」など、被害者意識をもったり、疑心暗鬼になったりしがちです。こんな状態にならないように、全員が気持ちよく協力できる体制を作ることも目的の一つになります。また、マルチベンダー環境では、他社ベンダーの協力が必要になりますので、お客様を通じて協力して貰える体制の構築も必要です。

(3) お客様の怒りを和らげる

障害が発生したことで、お客様は怒っています（あるいは呆れています）。原因がいずこにあれ被害を受けていることには変わりません。また、進まない対応に苛立っている場合もあるでしょう。まずはお客様の怒りを和らげて、今後の対策、対応等の説明を冷静に聞いてもらうことにより、共に協力して問題解決にあたる事が出来る様にします。

(4) 信頼の回復

システム障害を起こしたことにより、お客様は怒っているだけではありません。ソフトウェアの品質、障害調査のやり方等々、今までどおりで良いのか不安になっています。

この調査方法なら大事に至らずに問題は解決する。この見直し項目なら（今回と）類似の問題

は対策される。次回なにか起こったとしても、速やかに解決してくれる。今後も安心してソフトウェア（システム）を使ってもらえるようにすること、つまりお客様の信頼を回復することも重要な目的です。

（５） お客様への依頼事項をお願いする

暫定運用，継続調査，対策版ソフトウェアへの入れ替え等，今後の対応必要なお客様の協力をお願いすることも，障害報告書の目的の一つです。

１．２ ソフトウェア障害報告書の分類

ソフトウェア障害報告書の内容は，報告する時点での調査状況や，障害の原因が自社にあるのか？お客様側にあったのか？などによって注意すべき点が変わります。ここでは，ソフトウェア障害報告書を分類・整理し，それぞれの注意すべき点の違いを説明します。

（１） 報告する時点

（a） 第一報（初期報告）

発生した障害が特に重要なものや緊急度の高い場合には，連絡を受けた時点で直ぐにお客様に報告することになります。重要な障害でお客様も混乱している状況ですので，情報を共有し，協力して解決するための意識合わせを行い，お客様に落ち着いて頂くことが重要になります。マルチベンダーの場合でハードウェアや OS・ミドル等の他のベンダーの調査協力が必要な場合には，お客様を通じて依頼します。

第一報では次のような項目を報告します。

- ・現象，システム稼働状況，サービスの提供状況（確認のため）
- ・サービス復旧・調査の見通し
- ・必要に応じて資料の追加採取依頼
- ・関連する他社ベンダーへの調査依頼
- ・調査体制（窓口，責任者など）

調査の長期化やデータ修復の困難が懸念される場合には，それに応じた覚悟をして頂けるような説明をしておく必要があります。さらに，お客様が最終利用者や監督官庁などに通知・報告する必要がある場合には，その内容を判断するための情報を提供します。

（b） 中間報告

調査が長期化している場合には，途中で調査状況などを報告する必要があります。これは，「ちゃんと調査してくれているのだろうか？」とお客様の不安を解消するためにも重要です。中間報告では次のような内容を報告します。

- ・調査状況（推定原因，推定を確認するために調査しているところ，解決時期の目処等）
- ・情報収集版・毘がけ版ソフトウェアの適用依頼（必要な場合）
- ・追加資料の取得（必要な場合）

- ・運用回避策
等々

このままの調査方針で良いのかを意識合わせし、また、資料の採取が必要な場合には、協力を依頼します。

中間報告は定期的に行う場合と、調査に進展があった場合、情報収集版や暫定対策の適用依頼など調査の進展に応じて行う場合がありますが、調査の長期化が予想される場合には、中間報告の時期を事前に調整しておくことが望ましい。

※某企業でシステム障害が起こると、6時間毎に中間報告が要求されると言われていました。こうなると、調査と報告書の作成を分業で行う体制が必要です。

（c）原因判明時

障害調査により原因が判明した時に、対策、類似の不具合の見直し等を報告します。

- ・障害原因
- ・対策（必要に応じて運用回避策、暫定対策）
- ・類似の不具合の見直し策（観点、項目、期間）
- ・根本原因と再発防止策
- ・対策版ソフトウェアのリリース時期

この時、運用回避策の適用や、対策版ソフトウェアの入れ替えをして頂く必要がありますので、それをお願いすることも重要です。

（d）調査中断時

サービス再開を優先したため必要な資料が十分に採取できなかった、トラブルシュート機能の不備、障害発生時の資料採取手順の誤り等々により原因の究明が出来ない場合もあります。この場合は、現象が再現した時に必ず必要な資料が採取できるような仕掛けを作り調査を中断することがあります。この時には、次の項目を報告します。

- ・お詫び（原因が究明できないお詫び）
- ・調査状況（推定原因と原因を特定するために欠けている情報）
- ・必要な資料の採取策（情報収集版の適用、資料採取ツール、資料採取手順書等）
- ・現象再現時のサービスへの影響とサービス復旧方法（特に、データベース等の情報が失われること）

原因が究明できないまま調査を中断するのですから、そのことのお詫びが必要になります。また、障害の現象が再現した場合にサービスに与える影響が最小限になるような対策を考慮しておく必要があります。特に、データベースが破壊されるような障害では、データベースの回復策により失われるデータがないことを確認しておく必要があります。

（e）最終報告

原因に対する対策が済、類似の不具合の見直しが完了した時点で最終的な報告します。この場合、この報告書だけを見れば障害に関する全容が判るように記載します。

- ・現象
- ・直接原因，根本原因
- ・対策，再発防止策
- ・類似の不具合の見直し結果（観点，見直し項目数，摘出問題点）
- ・類似の不具合の対策版発行時期

また対策，再発防止策等容易な場合にはいきなり最終報告ということもありますし，対処が混乱し揉めた場合には始末書という形式になることもあります。

（2）原因の所在の違い

システム障害の原因にはソフトウェア側に問題があるものだけではなく，利用者側の運用によるものもあります。この原因の所在の違いにより，障害報告書の書き方を変える必要があります。

（a）自社側に原因がある場合

ソフトウェアのバグ，ドキュメントの不備等原因が自社側のみにあり，お客様（利用者）に一方的にご迷惑をお掛けしている場合です。この場合には，原因究明と対策・再発防止策が間違いなく行われていることを理解してもらい，今後も安心して使って頂くことが重要となります。また，自社に否があるところは素直に謝罪して怒りを和らげることも忘れてはなりません。

（b）顧客側に原因がある場合

明らかな設定ミスや操作ミスなどお客様側の問題により発生した障害の場合，そうは思っても一度立ち止まって，ミスの誘因が自社にないか再度検討して下さい。マニュアルや操作説明書などのドキュメントに不備はないか？ミスを抑止するためのインタフェースは十分だったか？エラーメッセージやその後の処理は妥当であったか？などを再度検討します。もし，自社側にもなんらかの問題があったら，一方的にお客様の原因と報告するのは危険があります。ここで，自社側の問題が発見されたばあには次項（「両者に問題がある場合」）を参照してください。

純粋にお客様側の原因である場合には，現象，原因，防止策（ミスをおかさないように注意すべき事項）等を簡潔に報告すればよいでしょう。この場合には，電話等口答の報告等で了解が得られ，障害報告書を必要としないこともあります。

（c）両者に問題がある場合

当該の障害がお客様側の操作ミスや設定ミス等に起因するのではあるが，ソフトウェアの作りが悪いためサービス与える影響が大きくなる場合があります（例えば，某証券会社の誤発注事件※）。この場合には報告の仕方（報告書の書き方）により，事後処理が大きく変わりますので細心

の注意が必要です。

この場合、利用者の使い方が悪いと断言し、ソフトウェア側の問題を認めない。というのは、技術者の心情的にはありがちですが、お客様から大きな反発を受け、ソフトウェア側の不備を責められることになり、対策費用・損害の負担等事後の処理で揉めるでしょう。その一方でソフトウェア側の不具合に焦点を当てると、その後の損害負担をどうするかという交渉に発展したときに不利になるのは明白ですお客様側の状態を十分に見極め、利用者の使い方が悪かったことによる責任とシステム側の不具合との間で落としどころを考えることになります。

お客様側のミスが一義的な原因で損害等の費用はお客様側に負担して頂き、自社側の負担で今後の再発防止のためにミスを抑止するようなインタフェースに改修する、というのが目指すべき落としどころでしょう。

※誤発注事件では、利用者の操作ミス（入力ミス、警告メッセージの無視）による誤発注が、発注取り消しが出来ないというソフトウェアの不具合により、損害が拡大してしまいました。さらにこの場合には、運用者による取引の停止が遅れるという運用ミスも重なり、損害賠償を求める裁判に発展しています。

（d）原因が判明しない場合（調査中断時）

原因が判明しないまま調査の中断をする場合には、どちらに原因があるかは明確になりません。この場合は、原因が究明できなかったことをお詫びする必要があります。（詳細は、前節参照）

（e）他社ベンダーに原因がある場合

マルチベンダー環境で他社ベンダーが担当するハードウェアやOS・ミドルソフト等が原因の場合には、当該ベンダーから報告することが普通ですが、その原因によりどのように現象が引き起こされるのかをアプリケーションの開発・保守者が報告することがあります。

この場合には、原因が現象に結びつくメカニズムを簡潔に説明すれば良いのですが、システムテスト・連動テスト等で何故事前に検証できなかったのか？が問われ、場合によっては、再発防止策として、関連製品との検証の手順を追加する必要があります。

※例えば、OSのパッチの不具合で問題が発生した場合には、パッチの検証手順を見直す必要があります。

1. 3 ソフトウェア障害報告書の読者

文書を作成する場合には、読み手を意識して書くのがビジネスの基本です。ソフトウェア障害報告書の場合には、この読み手の関係が複雑になります。そして、それぞれの読み手には立場があり、自分の立場を危うくする記載には寛容になれません。これが、ソフトウェア障害報告書を書くことの困難さを高める理由のひとつとなっています。

一般的な障害報告書の読み手は次の通りです。

- 社内の上司
- 社内関連部門
- 顧客担当者
- 顧客担当者の上長
- 関連他社ベンダー
- 顧客関連部門・監督官庁
- 裁判所

（１） 社内上長

まず始めに障害報告書を読むのは社内の上長、つまり、プロジェクトのリーダーやコンポーネントリーダーです。作成した報告書が品質保証部門や営業部門など社内他部門のレビューやお客様に提出にふさわしい品質にあるかどうかチェックされます。

せっかく書いた報告書も上司のレビューが通過しなければ、お客様に出ることはありません。報告書の構成、説明は判りやすさ、技術的な正確さ、無理な約束、誤字脱字等々、書き直しが多く発生するのもこのタイミングでしょう。

また、上長にも、直属の上長またその上長と階層がありますので、直属の上長の立場を悪くする（直属上長の評価が下がる）様なことを書くと、書き直しが命じられます。例えば、障害が起きた原因が直属上長のチェックが甘いことだったり、規則遵守が疎かであったりするような場合です。

とは言え、事実を変えることは出来ませんので、上長の責任に帰すような原因分析が出た場合には、事前に了解を得ておくなどの根回しが必要です。

（２） 社内関連部門

ソフトウェアの障害報告には、営業部門や品質保証部門が関係し、それぞれの立場からの指摘があります。勿論、社外に出す文書ですので様々な角度から見る必要がありますが、自部門の立場を守ることが優先されることもあり、厄介な場合があります。例えば、営業部門が、自社の利益よりもお客様との関係維持を優先することがあります（ので注意が必要です）。

また、問題があるソフトウェアがなぜ出荷されてしまったのか？途中でチェックできなかったのか？など、ソフトウェアのバグが原因の場合には、品質保証部門の反省も報告書に含めることがあります。

（３） 顧客担当者

障害の調査状況や今後の対応を、お客様の担当者に理解していただくことが報告書の重要な目的の一つです。お客様担当者のタイプに合わせて、記述内容、記述レベルなどを選択する必要があります。例えば、技術までちゃんと管理しようというお客様には、図を駆使してソフトウェアの動作を判りやすく説明する必要がありますし、技術にはあまり興味のなく再発防止を重視する

お客様もいます。

さらに、お客様の担当者にも上長がいますし、お客様内の関連部門や監督官庁があります。それらとの関係においてお客様担当者の立場を失うような内容を報告すると、会議は混乱します。

(4) 顧客担当者的上長

勿論、お客様の担当者にも上長がおり、その上長との関係において担当者の立場を気にする必要があります。例えば、自社側の対応が不十分なことが原因であっても、それを当然お客様側でチェックアウト出来る筈なのにしていなかったと、いった場合、相手側も立場を失う事になります。そのため、自分たちが悪くない場合でも、担当者の顔を立てつつ、管理者には自分たちは環境の範囲内で十二分に努力したことを伝えなければなりません。※

※ただし、お客様の担当者的対応が悪く、それをお客様の上長に伝えたい場合になど特別な事情により、揉めるのを覚悟で報告書に記載することも一つ的手段としてあります。(これには、問題の人を替えてもらうか、こちらが替えられるか、刺し違えるくらいの覚悟が必要です。)

(5) 関連他社ベンダー

マルチベンダー下の環境では、他社ベンダーにハードやOS・ミドル等の調査を依頼することがあります。この場合には、他社に調査を依頼する観点でインタフェースを中心に記載します。

例えば、他社部分に渡すパラメータや、戻されるリターンコードやエラー情報、ログを特定する日時などになります。

また、自社独自の技術を報告書上に記載してしまい他社に漏れることが無い様に注意することも必要です。

(6) 顧客関連部門・監督官庁

指導官庁が直接読むことはありませんが、お客様が報告しなければならないことがあります(例えば、銀行のオンラインが止まると、金融監督庁に報告が必要なように)。

こういった場合、お客様が報告書する元ネタになりますので、それを意識した形式で報告書を作成すると喜ばれます。もちろん、そのまま出るわけではありませんが、報告書を作成する場合にカット・アンド・ペーストで利用できると手助けになります。

(7) 裁判所(裁判官)

通常の場合裁判官が読むことはありませんが、システム障害が事件になり、裁判沙汰になることが稀にあります。その時には、障害報告書も重要な証拠物件として扱われますので、事前に用心しておく必要があります。これについては別途解説します。

2. ソフトウェア障害報告書の構成

(1) タイトル

報告の内容が判るように一言（一行）で書きます。末尾は「報告」、「中間報告」、「お願い」などで結ぶことが多い。

(2) 宛名

報告書の提出先（読んでもらう部署）を書きます。通常は会社名まで部署名をかくことは少ないでしょう。間に SI ベンダーが入っている場合等の様に直接のお客様と最終利用者が違う場合には、特に注意が必要です。このような場合には、どちらを記載するのか事前に直接のお客様に確認しましょう。

(3) 発行元・発行日

発行元には自社名（部門も併記することがある※）を記載することが多いのですが、直接のお客様と最終利用者が違う場合には、宛名と同じように事前確認をしましょう。

発行日は、通常報告日にします。

※個人名や個人の認め印を要求される場合もあります。事前に当該のお客様や営業部門に確認しましょう。

(4) 挨拶

ビジネス文書でよく使われる挨拶※ではじまり、報告する内容に関連した挨拶（お詫び）および報告の概要を記載します。報告する内容についての挨拶は「ご迷惑をおかけして申し訳ありません」、「心配をおかけしております」などのお詫びの意味を含んだ言葉がよく使われますが、これらの言葉を安易に使用することには危険があります。（詳細は「3章挨拶は細心の注意が必要」を参照）

※時候の挨拶や、「時下ますますご清栄のこととお慶び申し上げます。」ってやつですね。

(6) 現象

お客様からみたソフトウェア（システム）の現象を書きます。例えば、「×××プロセスが異常終了した。(×)」ではなく、「〇〇システムの□□サービスが△時間停止した。(〇)」と、お客様の視線で記載します。

(7) 調査の方針および状況（第一報や中間報告）

第一報や中間報告で原因が判明していない場合には調査の方針や状況を記載します。

- ・現状の調査で判明したところ、まだ判ってない所
- ・今後の調査方針

- ・追加資料の要否
- ・調査体制（連絡窓口，責任者，
等々

（８）技術的原因と 対策（原因が判明した場合）

技術的な原因とその対策を記載します。ここでの対策は，ソフトウェアに不具合の直し方，システムパラメータの設定変更，運用手順の変更等，ソフトウェアやシステムを変えるものになります。さらに，類似の不具合の見直し結果，およびそれに対する対策も記載します。

また，技術的な説明は判り易さを考慮し，図を使うべきでしょう。

（９）根本原因・再発防止策（原因が判明した場合）

根本原因を究明し，それにあわせた再発防止策を記載します。

根本原因は技術的原因と異なり，手順の漏れ，問題をチェックアウトできる仕掛けがなかった，担当者の教育や規則の遵守が徹底されていなかった等々，プロセスやプロジェクト管理の問題になります。そのため，再発防止策もそれに対応したものになります（「７．原因と整合性のとれた対策」参照）。

（１０）今後の日程

今後の障害の調査（中間報告の場合），対策の日程を報告します。

例えば

- ・原因調査用の情報収集版・罫がけ版の提供日
- ・パッチ版の提供日
- ・類似の不具合の見直し日程
- ・対策版の発行日
- ・次回報告予定日

等々

（１１）お願い事項

お客様の対応が必要なことをお願いします。

- ・回避策での暫定運用
- ・再発時の情報収集
- ・対策版への入れ替え
- ・システム運用上の注意（※誤操作等お客様側の誤った運用が原因であった場合） 等々

３． 挨拶は細心の注意が必要

技術者としては，障害の原因やなどの技術的な話をどうやって判ってもらえるかに注意を注ぎがちですが，「挨拶」を単なる飾りと軽視してはなりません。この「挨拶」によって報告の会議の流れや雰囲気が決まってしまうといっても過言ではありません。

対策に急を要し、切迫した雰囲気のお客様に、冗長な時候の挨拶から始めるのは怒りをかうかもしれません。「あんたらのお陰で大変な目にあってんだよ。それが判んないの」と思われたら、その時点で敗戦が決定してしまいます。営業部門を通してクレームが自社の幹部に届くことは必然。すると障害対応の難度が2段階はアップするでしょう※。

「ご迷惑をお掛けして申し訳ありません。」と、謝ってしまえば、お客様は落ち着くかというところ、それはそれで問題があります（詳細は「4. 謝罪について（何を謝るのか？）」参照）。

また、この時点で自社側の原因であることが判明している以外場合には、誤解を受けないように「不良」「バグ」等原因がベンダー側に存在することを示す言葉を使うのは避けましょう。調査の結果自社側に原因がないことが判明した場合に、誤解を取り除く苦勞を要します。原因が明確でない場合には、それに応じて「システムの不具合」「動作の不具合」等原因の所在を明示しない言葉を使います。

※人的資源などの支援は得られるのは助かるのですが、報告が増える、現場を指示ができるなどやりにくくなることも多いのですね...

4. 謝罪について（何を謝るのか？）

障害報告書に謝罪の文をどう載せるかどうかは、今後の人的資源、財務資源や損害の負担をどうするかを決める鍵となる項目です。

一方的に自社側の過失であれば謝罪の文章を入れるのが妥当ですが、ここで特に問題となるのは原因がお客様側にある場合、および、操作ミスや設定ミス等に起因するがシステム（ソフトウェア）の不具合で損害が拡大した場合です。

今までの日本的な商習慣では、まず「謝る」ことから入るのが原則でした。従って、報告をうけるお客様側は謝罪文が含まれていることを期待していることが多いのが実情です。お客様側が期待すると営業部門も謝罪文をいれることを主張することも、ままあることでしょう。

しかしながら、謝罪の文章報告書として残っていることは、話が拗れ損害賠償の交渉や裁判に進展した場合に不利な材料になります。未だに法律家は、ソフトウェア開発・保守の状況を理解することが出来ません。従って、証拠となる文書の表面的な言葉で判断することになります。つまり、謝罪の言葉があると自社側に否があると判断される有力な材料になってしまいます。過去には、いきなり電源ケーブルを引っこ抜くようなマニュアルに禁止が明記されていること操作が原因の障害であっても、障害報告書の謝罪の一言で裁判に負けた、というような事例が存在したとのことです。むやみに謝るべきではありません。

一方で謝罪文には話が拗れるのを防止する効果もありますので、状況は複雑です。確かに、軽微な損害であれば、「ご迷惑をお掛けして申し訳ありません。」と謝罪の文章により許され賠償を請求されない場合があることも事実です。

つまり謝ると不利になるし、謝らないと話がこじれて損害賠償問題に発展しやすい。では、一体どうしたら良いのでしょうか？

実は一義的な回答を示すことは出来ません。ケース・バイ・ケース。お客様のミスと自社側の不備との責任の度合い、お客様の状況、損害の大きさ等々を考慮して、日本的な善処を目指すのか、裁判も覚悟で徹底的に戦うのか？落としどころを探します。注意しないとならないのは、障害報告書を作成する前にしなければ

情報収集が重要になります。現地に、可能なルート全てを使って状況を把握します。そして落としどころを探します。

場合によっては法務部門に相談する必要があります（その様な規則の会社もある様です）。

とは言え、謝罪の文言をいれないとお客様が受け取ってくれなかったり、自社の営業からクレームがついたりする場合などがあります。思った様にシステムが使えなかった訳ですから、お客様は怒っています。お客様をなだめるために、営業部署等から「謝れ」との指示があるかも知れません。このようにどうしても謝罪の文章を入れざるを得ない場合もあります。この場合には何を謝るのが問題です。ソフトウェアの障害を起こしてしまったことを謝ってはなりません。もう一度、冷静にお客様がお怒りになっている理由を考えましょう。障害の連絡を受けてその後の対応に拙い点はなかったでしょうか？報告はタイムリーに出来たでしょうか？原因の判明までに予想以上の時間がかからなかったでしょうか？等々なにかある筈です。そして、その点の謝罪を報告書に記載するのです。

5. 顧客の調査依頼に合っている現象

現象の説明はお客様側の視点で、障害の調査依頼を頂いた時の表現に合わせます。我々技術者は、つついソフトウェアの内部で起こっている事象の本質と思われることを書いてしまいがちです。例えば、「あるプロセスが異常終了した。」「〇〇〇コマンドが△△△のエラーメッセージが出力された。」など。これらは、同じような不具合が起きた時、同じ現象かどうか判別するために役に立つという利点があります。

しかし、お客様から見るとこれらはどうでも良いことです。お客様から見えるのは、プロセスが異常終了した結果「サービスが停止した。」ことであり、コマンドがエラーメッセージを出したことにより「△△△の登録が出来なかった」ことです。そして、この表現で障害を連絡します。そのため、報告書の現象に「あるプロセスが異常終了した。」と記載すると「それってなんだっけ？」私がお願いした調査と違うんじゃないの？ということになります。

現象は調査依頼の表現に合わせるようにし、内部の動きは原因の中で説明するようにします。

6. 原因は図で示せ

障害報告書に限らずドキュメントから図が減ってきているように思われます。特に報告書をパソコンのワープロ・ソフトで書く場合に顕著なようです。手書きと比べると図が書きにくいのは確かなので、その影響だと思われます。

しかし、ソフトウェアの動きを文章だけで記述するのは困難です。正確をきそうとすると文章

はやたらと長くなりますし、そうなるを読んでいるほうも理解に頭を悩ます。報告にいった先でなかなか判って貰えず、挙句の果てにホワイトボードに図を書きだす。お客様からは「その図も入れてね。」と、いうことで、結局は報告書を再提出する、なんてことに身に覚えはないでしょうか？

確かに報告書の作成を急いでいる状況で一から図を書くのも非合理的ですので、事前に対策を考えておきましょう。機能仕様書等から流用するのも手段の一つです。また、障害報告書で使った図を蓄積・再利用するのも、報告書の品質向上と工数削減の一助です。

7. 原因と整合性がとれた対策

一般に公開されている障害報告を見ると、本当にこれで大丈夫なのと思われるものに出くわせることがままあります。それは、原因と対策の整合性が取れていないためです。

まず、直接原因と修正内容・類似の不具合の見直し、根本原因と再発防止策を対応させて記載しましょう。

さらに、原因とその対策との対応を判りやすくする必要があります。次の＜判りにくい例＞では、どの対策がどの原因に対応するかが判りにくいいため、正しい対策なのか？全ての原因に対策がとられているのかが一瞥できません。＜判り易い例1＞＜判り易い例2＞の様に、対策の末尾に対応する原因のインデックスを付与したり、表形式にしたりすることをお薦めします。

例)

＜判り難い例＞

●原因

(1) ○○○○○○○○

(2) △△△△△△△

(3) □□□□□□□

(4) ☆☆☆☆☆☆☆

●対策

(1) ●●●●●●●●

(2) ▲▲▲▲▲▲▲▲

(3) ★★★★★★★★★

原因と対策の対応がとれないため、「正しい対策がとられているのか?」、「全ての原因に対策がとられているのか?」が判り難い。

＜判り易い例1（文末参照）＞

●原因

(1) ○○○○○○○○

(2) □□□□□□□

(3) □□□□□□□

(4) ☆☆☆☆☆☆☆

●対策

- (1) ●●●●●●●● (原因 (1))
- (2) ▲▲▲▲▲▲▲▲ (原因 (2), (3))
- (3) ★★★★★★★★ (原因 (4))

<判り易い例 2>

#	原因	対策
1	○○○○○○○○	●●●●●●●●
2	△△△△△△△	▲▲▲▲▲▲▲▲
3	□□□□□□□	
4	△△△△△△△	★★★★★★★★

また、この対策で過剰な約束をしないことも重要です※。約束を沢山すればお客様は安心するかもしれませんが、そのぶんの原価がかかるだけでなく、人的資源がとられ予定していた他のことが出来なくなります。約束は必要十分に留める必要があります。

※類似の不具合の見直しの説明で、「え、そんなにできるの？」と、お客様に驚かれたという話を聞いたことがあります。

なにしろ報告した見直し箇所が数十万カ所。そんなのいつ誰がやるの、本当にできるの？これじゃお客様も期限を守って貰えるのか不安になりますよね。見直しがあまりにも少ないのもサボっているのかと疑問をもたれますが、多すぎるのも計画性があるのかと疑われます。せめて、ツールを使って機械的にやるから大丈夫等、見直しの計画が実現可能であることを示さねばなりません。

8. 重要なお願い事項

最後に、お客様にお願いする事項があれば記述を忘れてはなりません。不良を対策したソフトウェアへの入れ替え、運用上の注意事項（操作ミス等の場合）、現象再発時の資料収集等々お客様にお願いすべき事項は多々あります。これらの報告の席上口答でお願いするのではなく、障害報告書に載せてしまいます。情報収集手順や運用上の注意等量が多い場合には別紙にします。

9. おわりに

障害報告書を書くに当たり、技術的な内容を正確に分かりやすく書くことについては、かなり注意を向けられていると思います。一方で、ソフトウェア障害報告書のもつ政治的な面についてはあまり意識されていなかったのではないのでしょうか。本稿が障害報告書を書く上での新たな気づきになれば幸いです。

最後に、本稿作成に当たりDグループのメンバー各位の貴重な経験談を参考させて頂きました。この場を借りて深謝いたします。

(文責 高橋芳)

3. ま と め

第二十二年度活動を終えて ～幹事よりひとこと～

NTT データ CCS 馬場 辰男

企業活動に留まらず、産業や国家活動のほとんどが情報システムの支援によって遂行される時代、その情報システムの品質や保守の稚巧が、企業経営の命運を左右するばかりでなく、人命や国家の信頼にも決定的な影響を与える時代となり、ソフトウェアの保守は、運用を含めて科学的工学的に管理・対処されなければならなくなりました。

当研究会(SERC)は、このソフトウェアに付随する保守の問題を、開発標準、品質、用語、要員問題、プロセス、ツール等の種々様々な観点から研究してまいりました。また、その活動を通じて、ソフトウェア保守のJIS規格化にも少なからず貢献できたと自負しております。

私達の研究活動の積み重ねは、既に20年を越え、成果の蓄積は相当なものになっておりますが、一方、実社会での情報システムの運用は安定化するどころか、品質や保守の課題に起因すると思われるトラブルが、依然として多く報告されています。

私たちは、ソフトウェア保守に関心を持つ方々に、広くSERCを知っていただいて、共に切磋琢磨することで、自らの成長を期待すると共に、この活動が、参加者の皆さんのモチベーション改善と広く実社会のソフトウェア保守業務の改善に貢献できることを願っています。そして、いつの日か、日本の誇る技術として、ソフトウェア保守技術が海外に輸出されることを夢見ています。

東芝ソリューション(株) 増井 和也

今年2月20日、一般社団法人日本規格協会から1996年に発行されたSLCPのJIS規格の改訂版JIS X0160:2012 ソフトウェアライフサイクルプロセス(原国際規格:ISO/IEC12207:2008)が公開出版されました。その解説には「～ソフトウェア製品の供給、開発、運用、保守及び廃棄をするときに、適用。」と書かれていますが、規格自体を読むと、旧版にあった「開発プロセス」が無くなり、代わりに「実装プロセス」という名前になっています。

変更の背景は、ISO/IEC12207としてISO/IEC15288(システムライフサイクルプロセスの国際規格)と調和を取るためのようです。背景がどうあれ、今回の改訂で「ソフトウェア開発」という言葉が使われなくなったことはソフトウェア保守にとって一步前進だと私は考えています。

その理由は、「開発」の範囲の捕らえ方が曖昧であることが、ソフトウェア保守の範囲や保守プロセスの理解を妨げていたと私は考えていたからです。みなさんが「ソフトウェア開発」という言葉の使用をやめ「ソフトウェア実装」を使い、「実装」の範囲を正確に把握すれば、ソフトウェア保守の範囲や保守プロセスが自ずと正確に見えてくるだろうというのが私の見方です。

当研究会のみなさんには、ソフトウェアの保守や保守開発(改め保守実装?)の範囲やあるべきそれらのプロセスを考える際、今回のJIS X0160の改訂版(2012版)を参考にされることをお進めしたいと思います。

我々が使うソフトウェア保守という言葉の「保守」には、後ろ向きのイメージが付きまとい、クリエイティブさが感じられないとよく言われる。

本研究会もそれを踏まえて数年前に名称を「SERC」に変更した。保守を本来の持つ意味の「進化」に変えた訳である。しかし、イメージはそれほど向上していない。最近、情報システム学会の方とコンタクトする機会があり、そのメールマガジンで「アプリケーションの保守を考える」という論文を見つけた（参考文献を参照）。

そこには「maintenance には「よい状態を保つ」という意味があり、「リスクを恐れて変化しない」という意味ではない」と書かれている。さらに、日本語の保守という言葉は「後ろを向いて守る」というイメージを持たせてしまっている。これは「maintenance」を保守と訳したのは「誤訳」であった。「maintenance」は「維持」という言葉が妥当で、それは前向きであり、そのあとに「改善」という言葉が隠れている、という意味のことも書かれている。言葉の持つイメージは当に恐ろしいと感じた次第である。誤訳の呪文から抜け出すのは至難の技であるが、「maintenance」の本来の意味を正しく理解してもらうための活動も本研究会のタスクであろう。

・参考文献：情報システム学会 メールマガジン 2023. 6. 25 No08-03

企業及び社会生活における情報システムの意味を考える

第9回 アプリケーションの保守(maintenance)を考える

(<http://www.issj.net/mm/mm08/03/mm0803-kj-bt.pdf>)

著者：大島 正善 (Method Based Consulting)

(PS) この論文には他にも保守にまつわる示唆に富んだ事柄が多く
書かれて居るので是非一読をお奨めしたい。

アイエックス・ナレッジ(株) 田中 一夫(事務局)

今年度も、フォーラムを開催しようと試みたが、1回だけであった。猿でも反省するので、反省は置いて、次年度は、4回開催しましょう！会則も変更し、個人会員中心の会にしていきます。と事務局が云っても、幹事の皆さんや研究員の皆さんが納得して頂けなければ実現できませんが。